GAME DEVELOPER MAGAZINE

GAME OVER

UBM
Tech

ANIMATRIK
PERFORMANCE CAPTURE
ANIMATRIK FILM DESIGN • Vancouver, Canada
See why we are the triple-A choice at animatrik.com

001

CONTENTS_June/July 2013

VOLUME 20 NUMBER 06

game developer magazine

Postmortem

Features

Departments

REQUIEM FOR A ‘ZINE

GOODBYE, *GAME DEVELOPER*, AND THANK YOU

I've been dreading writing this column ever since I found out *Game Developer* was closing. Putting off the editorial until the last minute is standard practice for magazine editors, but this one has been extra tricky because, well, it's the last one ever. But in the end, after all the drafts are proofed, and the illustrations are placed, and the sacrificial offerings of coffee, cigarettes, and booze are made to the production and design folks, the words that run through my head are not "Goodbye," but "Thank you." The rest of this issue is about saying goodbye (we even made you a mixtape), so I'm going to use this page to express gratitude, instead.

TO THE READERS

First off: Thanks for reading. *Game Developer* wouldn't have lasted 19 years if we didn't have people like you out there who stuck around for issue after issue, devouring all the game development knowledge we could possibly throw at you. I'm convinced that our readers are some of the smartest and nicest folks I've met in the industry thus far, and I hope to see you among Gamasutra blogs and GDC talks as they try to pick up where we left off.

Since *Game Developer* is meant for professionals, we've had a relationship to our readers that I've never been a part of before—because you are often our writers as well. Other publications typically demand that the editorial staff function as capital-E Experts that pass knowledge down to the readers in a one-way relationship, but with *Game Developer*, my job has more or less been to provide readers with ways to share knowledge with each other. Some of the best stuff I've had the privilege to include in this magazine in the last year and a half began because a reader insisted that we should be covering X topic, or disagreeing with Y column—to which I responded, "You wanna write about it for us?" So thank you, readers, for helping us make the best damn game dev magazine we could possibly make.

TO THE EDITORS PAST

Game Developer has built up an excellent reputation over the years, and I credit that entirely to the dozens of talented people who tended to these pages before I did. I have been consistently surprised by the esteem with which industry people regard *Game Developer.* It felt humbling and inspiring to know that the work we did was due in large part to the industry giants on whose shoulders we stood. (It also made my day-to-day job a whole lot easier.)

TO THE BEHIND-THE-SCENES STAFF

There are plenty of people involved in making *Game Developer* that don't get any love besides a brief mention on the masthead. Thanks to the sales team for doing their damnedest to keep print alive; thanks to our wonderful advisory board for offering their time, knowledge, and expertise; thanks to the Gamasutra staff for their support and collaboration; thanks to production manager Dan Mallory for talking me down each time I was sure we weren't going to ship an issue out on time (this was usually about halfway through the third week of each issue cycle); thanks to art director Joe Mitch for busting his butt every month to turn out an amazing magazine in about a week; thanks to our copyeditors, Alexandra Hall and Carrie Shepherd, for lending me two pairs of eyes much sharper than mine. And while I'm at it, I'd like to extend a personal thanks to Simon Carless, Pearl Verzosa, and former EIC Brandon Sheffield for taking a chance on a new editor and giving me a turn behind the wheel. It was fun while it lasted.

INSERT CREDIT TO CONTINUE

If there's one thing I've learned from the last year and change working on the mag, it's that the people who read, write, and produce *Game Developer* do great things. There's an ex-*Game Developer* editor working in the White House now, for crying out loud—that's pretty cool! So I want you to read the rest of this issue cover-to-cover a few times—maybe put it somewhere safe and re-read it every year or so, because it's just that good—and rest assured that we plan on doing great things, and we want you to do so, too. Personally, I'll be sticking around as GDC's director of online community, so don't think you've seen the last of me.

In closing, I'll borrow a few words from author Garrison Keillor: Be well, make good games, and stay in touch.

The end!

Patrick Miller
Editor, *Game Developer*
@pattheflip

gd
GAME DEVELOPER MAGAZINE
WWW.GDMAG.COM

UBM LLC.
303 Second Street, Suite 900, South Tower
San Francisco, CA 94107
t: 415.947.6000 f: 415.947.6090

SUBSCRIPTION SERVICES

FOR INFORMATION, ORDER QUESTIONS, AND ADDRESS CHANGES
t: 800.250.2429 f: 847.763.9606
gamedeveloper@halldata.com
www.gdmag.com/contactus

EDITORIAL

PUBLISHER
Simon Carless *scarless@gdmag.com*
EDITOR
Patrick Miller *pmiller@gdmag.com*
EDITOR EMERITUS
Brandon Sheffield *bsheffield@gdmag.com*
MANAGER, PRODUCTION
Dan Mallory *dmallory@gdmag.com*
ART DIRECTOR
Joseph Mitch *jmitch@gdmag.com*
CONTRIBUTING WRITERS
Kris Graft, Christian Nutt, Frank Cifaldi, Brandon Sheffield, Mike De La Flor, Mike Acton, Steve Theodore, Jason VandenBerghe, Damian Kastbauer, David Edery, Kim Pallister, Matthew Wasteland, Magnus Underland
ADVISORY BOARD
Mick West Independent
Brad Bulkley Microsoft
Clinton Keith Independent
Bijan Forutanpour Qualcomm
Mark DeLoura Independent
Carey Chico Independent
Mike Acton Insomniac
Brenda Romero Loot Drop

ADVERTISING SALES

VICE PRESIDENT, SALES
Aaron Murawski *aaron.murawski@ubm.com*
t: 415.947.6227
MEDIA ACCOUNT MANAGER
Jennifer Sulik *jennifer.sulik@ubm.com*
t: 415.947.6227
GLOBAL ACCOUNT MANAGER, RECRUITMENT
Gina Gross *gina.gross@ubm.com*
t: 415.947.6241

ADVERTISING PRODUCTION

PRODUCTION MANAGER
Robert Steigleider *robert.steigleider@ubm.com*
t: 516-562-5134

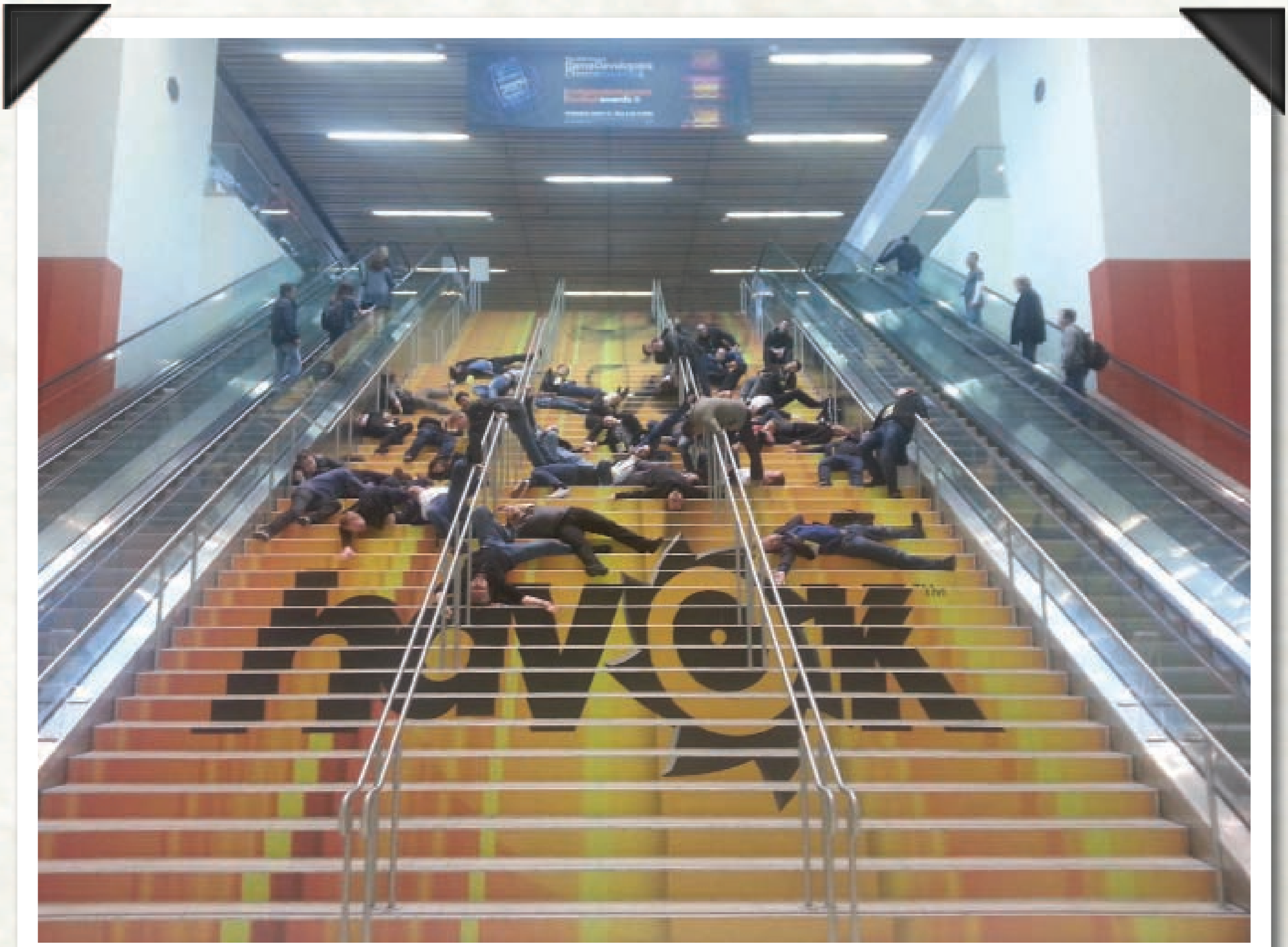
REPRINTS

WRIGHT'S MEDIA
Jason Pampell *jpampell@wrightsmedia.com*
t: 877-562-5972

AUDIENCE DEVELOPMENT

AUDIENCE DEVELOPMENT MANAGER
Nancy Grant e: *nancy.grant@ubm.com*
LIST RENTAL
Peter Candito
Specialist Marketing Services
t: 631-787-3008 x 3020
petercan@SMS-Inc.com
ubm.sms-inc.com

UBM Tech
WWW.UBM.COM


havok

GOODBYE, GAME DEVELOPER

Kind words and well wishes from *Game Developer* readers

Shortly after the news broke of *Game Developer*'s closing, we found ourselves awash in condolences, memories, and well wishes from our reader community. We wanted to preserve a few of these in print for posterity.

" Bummed out that after years of reading *Game Developer* and seeing colleagues published, the doors are closing. "

—Evan Groenke

" Thanks to Simon Carless, Brandon Sheffield, Patrick Miller, and all past and present *Game Developer* staff for delivering a killer industry publication. "

—Garth DeAngelis

" This magazine helped me find my way into game dev. Thank you for everything!"

—Brandon Perkins

" Great articles, oodles of eye candy, and good lessons learned. "

—Brian Zubert

" In my first game dev job, my boss suggested in a one-on-one that I read *Game Developer* for professional development, and I have ever since. :(RIP. "

—Shay Pierce

" One of only a few sources of institutional memory for game development is gone. RIP *Game Developer*. "

—Casey O'Donnell

" Thanks, *Game Developer*, for inspiring me to expand my understanding of the game biz. You will be missed! "

—Michael Marzola

" RIP *Game Developer.* It's a bitter thing to outlive your magazines. "

—Larry O'Brien, Game Developer founding editor

" Getting an article in *Game Developer* was a high point for me. "

—Megan Fox

" Oh my God, *Game Developer* closing is honestly heartbreaking. "

—Nels Anderson

" Wow. Very sad news. I started in the industry at the same time as the magazine started publishing and it was my best source of information. I read every page (sometimes twice). "

—-Clinton Keith

" Get back issues of *Game Developer* if you can. You'll learn a lot even if you don't fully understand. "

—Mitch Dyer

" I'm about to choke up in a meeting thinking about *Game Developer* folding. "

—Chris Charla

" RIP *Game Developer*, your *Deadly Premonition* cover story still fills my heart with love. "

—Shane Bettenhausen

" Oh no...you will be missed, *Game Developer.* "

—Craig "superbrothers" Adams

" About a week ago, my wife wanted me to clear some space on the counter and she pointed to a stack of *Game Developer* issues from about five years ago. So I thought, okay...I'll get rid of some of these older issues that aren't relevant anymore. Oh, yeah—heat-mapping the deaths of the players to see problematic zones in the levels—that was clever... Darn. So, I had to find somewhere else to put them, because in one respect or another, this retro data format remains relevant, at least to me. "

—Steven Ehrensperger

" WHAT WILL I READ WHILE ON THE TOILET?!?! "

—Ryan Barrett

GDMIXTAPE

LISTEN ALONG TO THE LAST ISSUE WITH OUR OFFICIAL PLAYLIST

Side A:

Icarus (DEUS EX: HUMAN REVOLUTION - Michael McCann)
E1M1 at Doom's Gate (DOOM - Bobby Prince)
Xenon (XENON - Ryu Umemoto)
City of Radiant Ruin (ETRIAN ODYSSEY 4 - Yuzo Koshiro)
To Make the End of Battle (YS BOOK I&II - Yuzo Koshiro)
Gene's Rock-A-Bye (GOD HAND - Masafumi Takada and Jun Fukuda)
Pissing on Your Grave (The Rites)

Side B:

The Fear Theme (Kenji Eno, The Cinematic Orchestra)
The Dream is Dead (Life is Killing Me - Type O Negative)
The Wretched Automatons (NIER, Monaca)
Passage from Genesis (ECCO THE DOLPHIN: DEFENDER OF THE FUTURE, Tim Follin)
Lonely Rolling Star (KATAMARI DAMACY, Yoshihito Yano, Saki Kabata)
Cloud 8 (Ray - Frazier Chorus)
Human (Crash - Human League)
Let's Go Away (DAYTONA USA - Takenobu Mitsuyoshi)

Dear Game Developer readers,

Wow, 19 years, huh. That's a long time. We've been through a lot together. And we've learned so much about each other. Um, this is really tough, readers, but there's something we have to tell you.

You've been great to us. You really have. And it's not your fault—honest!—but it's time for us to move on. Without you.

Oh—hey. It's okay. Don't cry. Er, it's okay to cry. Go ahead. Just *leeeet* it out. There, there. Shhh.

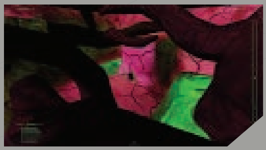
This isn't your fault. It's not you, it's us. The whole print magazine thing, it's just not working. It's not sustainable. No, really—what with the trees and all.

Um, we've got to go now. But we made you something to go along with our last issue—it's a mixtape. You can find it over here (http://bit.ly/YHnbiW). You'll love it, we promise. Chin up, now. What was that line from that Robyn song? "The only way your heart will mend, is when you learn to love again, and it doesn't make sense right now, but we're still your friend."

Hang in there. Maybe we'll see you, you know, later.

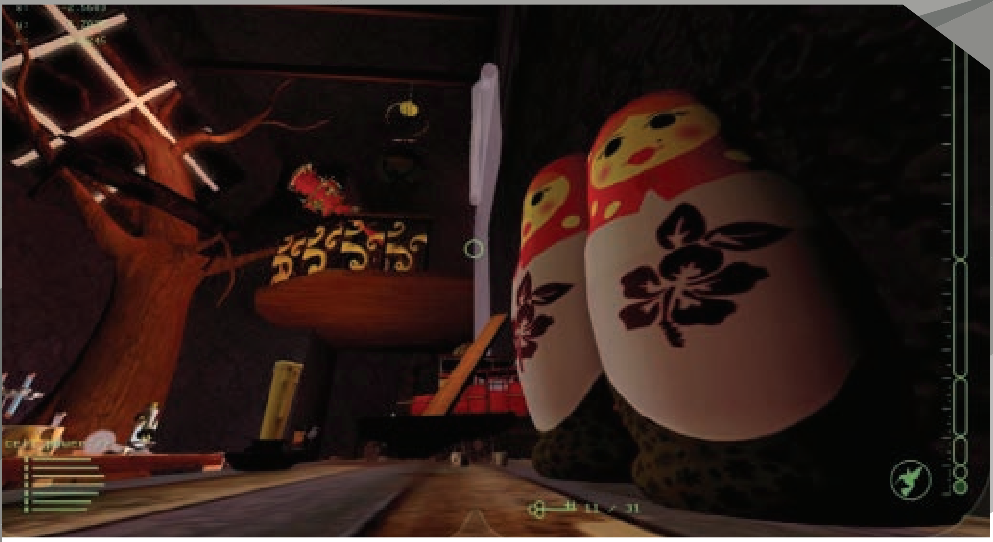
Love,

Game Developer Staff



TALES FROM THE MINUS LAB

WWW.THEMINUSLAB.COM



Developer: USC Games (University of Southern California School of Cinematic Arts Interactive Media and Games Program, Viterbi School of Engineering, collaboration with Atlantic University College)
Release date: TBD (current build available on theminuslab.com)
Development time: About a year
Development budget: N/A (student project)
# of lines of code in the game: unknown
A fun fact: When you're full size, the lab is about 30 feet across; at your smallest size, it is about 1.6 miles across from your perspective.

From the University of Southern California comes TALES FROM THE MINUS LAB, a first-person exploration game (and IGF 2013 Student Showcase honorable mention) created by a surprisingly large student team. The game lets you change size, which dramatically affects your experience exploring a single, very detailed room.

ALEXANDRA HALL: How'd you come up with the shrinking/growing concept?
ALEX BEACHUM: I was working on a quick prototype for a class a few years ago. The assignment was to create some form of "navigable space," and I was curious how the ability to shrink would affect your perception of what appeared to be a pretty confined environment (in this case a tiny prison cell). That concept of using shrinking/growing to explore the spaces-within-spaces nested within a single room is still a huge part of the game.

My biggest initial source of inspiration was probably *Honey, I Shrunk the Kids* (probably due to growing up in the '90s); in TALES FROM THE MINUS LAB we tried to capture that sense of exploring a vast wilderness that's always been there underfoot, you just never looked close enough to notice it. THE LEGEND OF ZELDA: THE MINISH CAP was also an inspiration.

ALEXANDRA HALL: Why did you write your own engine?
ALEX BEACHUM: We were fortunate enough to be working with two very talented lead engineers (Steve Wenzke and David Young), who determined that rolling our own engine would give us complete control over the complicated physics we knew the game would need. We did indeed gain complete control over physics, and the programmers on the team got fantastic experience writing an engine in C++. [But] it was difficult to iterate rapidly due to the lengthy process of setting up environments and content in the engine.

ALEXANDRA HALL: At 38 people, your team is quite large. Were there any management challenges?
SARAH SCIALLI: This was definitely a huge management challenge. It required us to be more formal and hierarchical than we had initially expected. We had to divide into smaller specialized subgroups, for example, a small user interface team of a designer, artist, and programmer that set their own internal deadlines and reported back. On the whole, the large team made keeping everyone informed between specializations much harder.

ALEXANDRA HALL: Did the massive team end up being an asset?
SARAH SCIALLI: In some ways [it was] more of a hindrance than a help. Though it was great to have so many talented individuals working on the project, it incurred a lot of management overhead to keep track of the sheer number of tasks happening at once. And, because each person was focused on their own specific tasks, it was harder for each person to understand how each task fit into the bigger picture. However, it was great that team members were able to become experts in their specializations.

ALEXANDRA HALL: Is it tricky to design environments that support gameplay at multiple scales?
ALEX BEACHUM: That was the most difficult design challenge of the entire project. We wanted to create situations where players would have to shrink/grow fairly often, so we spent a lot of time early on thinking about the pros and cons of being different sizes. For example, being big means you can pick up and manipulate objects (even entire levels), move across the lab quickly, and overall you're less vulnerable to hazards in the environment. Being small allows you to explore tiny nooks and even walk on water (surface tension!), but falling off a table is roughly equivalent to falling off the top of a small mountain.

Something that's easy to forget is that, from the player's perspective, it's as if the space itself is shrinking/growing around you. An environment that feels really expansive at your smallest size (about the size of an ant—or 256 times smaller than full size) can literally fit inside a beaker perched atop a shelf in the corner. Since the entire game takes place in a single room, level design is all about stitching together these recursive spaces in a way that feels organic and takes advantage of players' abilities at various sizes. It's like the Russian doll approach to world design!

ALEXANDRA HALL: What's next? Do you plan to continue development?
SARAH SCIALLI: We love TALES FROM THE MINUS LAB and hope to one day complete it and get it out to the world. A large chunk of the team has graduated and now have jobs in the industry. Alex and I are leading another Advanced Game Project this year: OUTER WILDS, Alex's MFA thesis. I'm also working on my MFA thesis, AGLAEA.
ALEX BEACHUM: Yep, I'm currently wrapping up development on my master's thesis (and advanced game project) OUTER WILDS. I still hope I get a chance to finish MINUS LAB though!

LET'S BE FRANK

GAMASUTRA NEWS DIRECTOR FRANK CIFALDI EXPLORES OTHER OCEANS

Veteran game journalist (and official *Game Developer* brunch buddy) Frank Cifaldi has been around the block. *Game Developer* readers will recognize him as sister site Gamasutra's former news director and occasional *Game Developer* contributor, and some might be familiar with his game-preservation work at Lost Levels (lostlevels.org). Recently, Cifaldi moved over to dev studio Other Ocean Interactive, so we wanted to check in and see how this card-carrying member of the press corps was faring on the other side of the fence.

Patrick Miller: Frank, we know you best as the former news director for Gamasutra. What, exactly, is your new gig?

Frank Cifaldi: I'm working with my friends at Other Ocean in beautiful Oakland, California now. My title is "designer," but so far I'm mostly doing biz dev stuff. We're a small enough team that I get to get my hands dirty with a little of everything, which is how I like it.

PM: What's it like to be on the dev side of things for a change? Has your perspective on the industry changed at all?

FC: I can't say there were any surprises coming over here, having covered the industry for something like 10 years now. (Ask me again after our first big crunch...)

As for what it's like: I think it's great. I've spent so many years turning around quick articles on the Internet that slowing down a little and working on a big project piece by piece is like a brain vacation!

PM: Has the industry expertise you've developed on the journalism side of things translated well to actual development? What kinds of skills or knowledge have crossed over?

FC: Content production is content production, as far as I'm concerned. Being able to identify what satisfies players and keeps them playing isn't much different from identifying reader trends and keeping them coming back to a website. Also, as with any job, being able to compose a proper sentence is always going to come in handy.

PM: You're also known as something of a game historian, of sorts. Has that come in handy so far? (Your encyclopedic knowledge of bad licensed games, perhaps?)

FC: That's why Other Ocean was such a good fit for me—we're a group of like-minded enthusiasts of the entire spectrum of video game history. (My desk is nestled tightly between a ROBOTRON and a TEMPEST machine, for crying out loud!) But yeah, we're able to communicate ideas to each other really easily because of how many old games we've all played. It's like a foreign language only the real nerds ever bothered to learn. Like Latin, except even less useful.

PM: At what point do you think Other Ocean will realize that all your pitches originated from your most recent* GAME DEV STORY *playthrough?

FC: I don't care what my failure in GAME DEV STORY says. WWII KARTS is a great idea for a game.

WHERE ARE THEY NOW?

GAME DEVELOPER'S EDITORS-IN-CHIEF

Game Developer founding editor **Larry O'Brien** is a developer evangelist for Xamarin, a cross-platform IDE meant for mobile development.

Mark DeLoura currently works at the White House Office of Science and Technology Policy as their senior adviser for digital media.

Alex Dunne still works for *Game Developer*'s parent company, UBM Tech, as the general manager of online products.

Jennifer Olsen (now Jennifer Yeamans) is a transportation planner for the Metropolitan Transportation Commission. (Sounds like real-life SIMCITY to us.)

Simon Carless oversees the Game Network and Black Hat subdivision of UBM Tech, which includes *Game Developer* and Gamasutra, as well as Game Developers Conference and infosec conference Black Hat.

Jamil Moledina is co-founder and CEO of a San Francisco-based dev studio named Wormhole Games.

Alex Handy is a senior editor for *Software Development Times*.

Brandon Sheffield is the founder of dev studio Necrosoft Games, and a senior contributing editor for Gamasutra.

Current *Game Developer* editor-in-chief **Patrick Miller** will be moving into a new role with the Game Developers Conference, as the director of online community.

* * *

Put on your game brain.

PopCap®

jobs.popcap.com

QUALITY OF LIFE FOR BETTER OR WORSE
INTERVIEW: SIGHTS CREATOR NAOTO OHSHIMA
gamedeveloper
THE LEADING GAME INDUSTRY MAGAZINE
DEVELOPERS! DEVELOPERS! DEVELOPERS!
PREMONITION
RESEARCH REPORT: GOVERNMENT INCENTIVES
flawed beauty
GUITAR HERO
POSTMORTEM MIDWAY'S JOHN WOO PRESENTS STRANGLEHOLD
PRODUCT REVIEWS
INTROVERSION'S DEFCON
THINKING WITH PORTALS DEVELOPING VALVE'S NEW IP
MASSIVE ATTACK
HOT METRICS
SINGULARITY
FEATURE: TOP 50 DEVELOPERS REPORT
tomb raider underworld
truly massive
STACKLESS PYTHON FOR CONSOLES
TOOL BOX MOTIONBUILDER 2009 REVIEW
VISUAL LEARNING INTRODUCING TOOLS TO YOUR ART TEAM
STUBBS THE ZOMBIE
THE ZEN OF SHIGERU MIYAMOTO
THE FIRST ANNUAL BUSINESS ISSUE
EA SPORTS FIGHT NIGHT 2004
2008 FRONT LINE AWARDS ARE HERE!
MEDIA MOLECULE'S LITTLE BIG PLANET
A COLLECTIVE BACKBONE
DOUBLE FINE'S PSYCHONAUTIC BREAK
PRODUCT REVIEW MAYA 7 UNLIMITED FOR MAC OS X
SPECIAL FEATURE: TOP DEVELOPERS OF 2008
the conduit postmortem
UNCHARTED 2 AMONG THIEVES
PROGRAMMERS STRIKE BACK
PRODUCT REVIEWS SIDE EFFECTS HOUDINI 9
2K GAMES'

Postmortem: Game Developer Magazine

THE UPS AND DOWNS OF A 19-YEAR-OLD GAME DEV MAG

By Brandon Sheffield

PRODUCT REVIEWS ENDORPHIN 2 CLAYTOOLS SYSTEM
game developer
POSTMORTEM: DOUBLE FINE'S PSYCHONAUTIC BREAK

AUGUST 2005

MAY 2008

DECEMBER 2009

Game Developer has had a good run. We started in March 1994 as a quarterly, and moved quickly into a monthly publication as demand grew. Over the years we've seen over two dozen editors, multiple company name changes, four major design overhauls (frankly, there should've been more), and hundreds of articles that have helped developers do their jobs better (we hope).

And now, a few months shy of its 20th birthday, after winning a slew of magazine industry awards, *Game Developer* is coming to a close, because print is no longer an attractive market. There was a lot more we wanted to do, and we hoped to be able to serve you for years to come. The magazine was taking more of an indie and small-team bent under Patrick Miller's leadership as the market has shifted in that direction, and we hoped to open a venue for a host of new voices. As the magazine's longest-running editor, at eight years, I thought it might be fitting to do a postmortem of the entire operation in this, its final issue. Plus, Patrick and I figured that since we had been asking devs to take a frank look at what went right and wrong with their work for so long, it was only fair that we do the same with ours.

What Went Right

1. THE "PRESTIGE" OF PRINT

People respected the magazine, which certainly made us editors happy. Often at trade shows we would get comments like "Oh, *Game Developer*! That's the one publication I read!" Granted, most of these people were getting the magazine for free, but it was clear that people viewed it as respectable (even though most folks outside the U.S. had never heard of it).

No offense is meant to our Gamasutra siblings, but we would often get pitches saying, "Well, I guess I could put this on Gamasutra, but I'd really like to have it in print." For some people, print still had that allure of being "published," and having written something of import. It's almost analogous to the prestige of making a console game—we grew up with it, so it must be "better," right? This is why so many game developers have blindly put their games on consoles, when other platforms are doing much better. We like history, I suppose!

That went for us editors, too. Every month we would pour ourselves into this thing, staying up late, working overtime, putting final touches and making last-minute edits to ensure the magazine was the best it could be—and at the end of our monthly cycle, we'd have an actual product we could touch. It's a great feeling. Many game developers work for years on a game, and when it's out, they have to start all over again. We got to do this monthly, knowing we were actually creating something.

2. TIES WITH GDC

As much as I personally wanted to differentiate the magazine from its parent company, I will admit that our ties with GDC were helpful. We got greater visibility at the shows, not to mention a generous ad boost, and it didn't hurt that the higher-ups looked at the magazine as a sort of soft marketing arm for the show. It helped us maintain relevance and ground to stand on as other magazines closed around us. That couldn't last forever, but it helped for a time.

In fact, I think a greater convergence would've been helpful—turn Gamasutra into gamedeveloper.com, and unite all the properties. But, alas, we will never know!

3. ANNUAL STATE-OF-THE-INDUSTRY REPORTS

For over a dozen years the magazine had been running reports like the Salary Survey and the Front Line Awards, which are the only contiguous running reports on game developer salaries and ranking of game development software—period.

Neither of these was perfect, we'll be the first to admit, but there wasn't much better to be done. As my former managing editor Jill Duffy said to me, "We did that shit right by hiring an outside expert to conduct the survey and analyze the results, and it was smart to stick with the same contracted partner year over year."

Since statistics were outside our wheelhouse, for the Salary Survey we used a professional (and expensive!) statistician to sift through thousands of survey responses to bring us valuable data. Some folks have criticized the survey, because they feel the numbers are inflated—but it's really the best we could have done without somehow getting all dev studios to give us their employee numbers. When that's the sort of industry we work in, we'll all be holding hands and hugging rainbows, and won't need money anyway!

The Front Line Awards are some of the biggest industry awards on the game development software side, and companies actually cared about these things. Epic, for example, has been touting its running tally as best engine for a number of years.

We've had a few other surveys and objective reports throughout the years, and have always done our utmost to make sure they were accurate. A lot goes on behind the scenes with these things, and I think we did quite well with them.

4. PROFESSIONAL, HARDCORE STAFF

This is tooting our own horn a bit, sure, and we've edited a whole lot of horn-tooting out of postmortems in the past—but it's our last issue, so begrudge us this. Our editors were pretty hardcore. We stuck to a style guide, did multiple edits and multiple art passes of every article, and generally tried to make the best magazine possible. One doesn't think a lot about the layout of printed words on a page and how they're juxtaposed to images until one has to lay out a feature with more than 30 figures and code listings.

The magazine is currently done in large part by two people: editor Patrick Miller and production manager Dan Mallory. We have a part-time art director in Joe Mitch, a part-time sales lead in Jennifer Sulik, and up until very recently, we had Pete Scibilia in New York to interface with the printers. That's it. We have contributors, sure, but it's a really lean operation. Only two people are working on the magazine full-time, and that's how it's been since around 2005. I sometimes think it's a miracle that a 52- to 96-page issue of any quality gets pumped out every month, but there it is.

We've had many editors over the years, but most of them didn't work together. Through hard work and a dedication to the craft, we managed to create a magazine that people liked, with what amounts to a skeleton crew.

5. FRANK, HONEST TONE

Many, many folks out there were more than happy to write a postmortem for the magazine ("That means we're on the cover, right?")—until it came to the What Went Wrong portion. But we took it as a point of pride that we were able to offer developers a chance to be frank and honest about their work and the industry, even if it meant that every so often we'd find out that we couldn't publish a postmortem we had spent weeks editing and revising with the writer because "crucial stakeholders still needed buy-in."

The game industry has precious few places where developers are encouraged to be honest with each other, and *Game Developer* was one of them. We truly believe that this is one of the best ways to help fellow devs make better games and advance the medium overall, and we wish Gamasutra the best of luck in carrying this tradition in the future.

To be sure, it's one thing to expect our contributors to write frankly about their professional shortcomings, and another to do it yourself. We fully acknowledge that *Game Developer* stood as strong as it did due only to the brave devs who were willing to bare their souls in print for the sake of helping others learn from their mistakes. After all, we couldn't have done anything without the hard work of 19 years of contributors and columnists, so ultimately, anything we've accomplished over the magazine's run is your accomplishment as well. Thanks, everyone.

What Went Wrong

1. INDUSTRY CONTRACTION MEANS FEWER ADS

Game Developer magazine was (mostly) free to qualifying customers, and like most business-to-business magazines, we relied on advertisers for revenue. Because of this, industry contraction hit us hard, throughout my tenure at the magazine.

First, we saw consolidation in the game software space. There used to be a whole slew of tools that advertised with

GDC:06 PREVIEW GUIDE INCLUDED
gamedeveloper
THE LEADING GAME INDUSTRY MAGAZINE
STUBBS THE ZOMBIE

JANUARY 2006

INTERVIEW: NIGHTS CREATOR NAOTO OHSHIMA
gamedeveloper
DEVELOPERS! DEVELOPERS! DEVELOPERS!
GAMING ON THE WIRE
scribblenauts

NOVEMBER 2009

PRODUCT REVIEW NEXTENGINE 3D SCANNER
gamedeveloper
THE LEADING GAME INDUSTRY MAGAZINE
METANET'S
N+

SEPTEMBER 2008

us—then, one by one, they started to get swallowed up by one company or another. This turned a hydra of advertisers into just a few sales points. They could advertise one product or another in a different issue, but since they were no longer competing, they didn't advertise in the same issue. This drastically reduced the number of advertisers in our pages.

Next to go was recruitment. Advertisers found it more valuable to advertise in Gamasutra, which had a wider reach and could be shared via the internet. They have moved even further into direct recruitment through Twitter and the like.

Couple this with the fact that from at least 2004 through much of 2011, we had nobody specifically assigned to do sales for the magazine. The emphasis was on GDC, which is obviously much bigger than the mag, and package deals. At one point some sales staff were making package deals with GDC and Gamasutra, getting multiple pages in the magazine as part of the deal. Digging hard through the corporate structure, I found that in one instance we actually sold magazine ads for less than it cost to print the pages they were on. Suffice it to say, that never happened again, but these deals weren't exactly giving us blockbuster profits.

Just as I was leaving, we got Jennifer Sulik in sales, who did a great job of bringing in new advertisers (from smartphone makers to auto companies), and getting our numbers stable. In fact, the magazine was on target to meet its profit estimations for this year. If we had gotten a dedicated salesperson earlier in our life cycle, I can't help but wonder where we'd be now.

This may surprise people to hear about a magazine that's closing down, but *Game Developer* was always profitable. Maybe not every month, but it was profitable every year of operation (so far as I know). To be certain, our contribution to our parent company (that is to say, our profit) was declining, but it was still profit! We kept lean, kept regular advertisers, and managed to squeak through a bit of digital revenue. You couldn't really do much better as a small-circulation magazine, but it wasn't good enough to keep our parent company's skin in the game.

2. DIGITAL: TOO LITTLE, TOO LATE

We rolled out our digital edition last year, and it was quite successful. But we should've done it years ago. Frankly, our infrastructure was so scattered, and our crew so small, that nobody had time to make it happen. We were too busy trying to get through the next day.

Many magazines were switching to a more digital-oriented model and trying to get actual paying subscribers, but we began the transition too late, and our digital model was imperfect at best. Many subscribers mentioned that they had a better experience by downloading PDFs of the magazines and reading it in a PDF viewer than they did with our actual app.

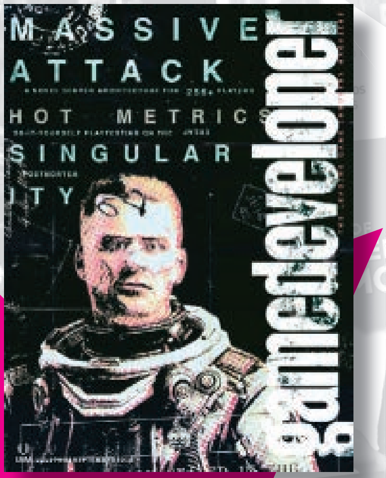
3. LACK OF INTERNAL SUPPORT

As has been hinted a few times in this postmortem, we were a bit starved for support. The magazine was chugging along at an even pace, which was great in some ways, because we didn't get a lot of scrutiny from the higher-ups. They were content to just let us do our thing. But at the same time, we also didn't have much support when we wanted to do something new—like launch a mobile edition, get new advertisers, reach new markets, and so on.

In early 2012, we finally got someone on staff—Pearl Verzosa—to tackle this sort of thing, but at this point in the magazine's life cycle, it was viewed as extra overhead, rather than an overhaul of a valuable piece of the business.

I mentioned that we had to share sales staff for many years, but sharing art staff was difficult as well. I went through a few art directors in my time, and Joe Mitch has been a great one—but he was often pressed for time, and had to squeeze the entire look and feel of the magazine into four or five days' work. In fact, that's all he was budgeted for. This could lead to bottlenecks that would have a ripple effect through production.

But through hiring a professional production editor (instead of simply training one, as we'd done for years), and having a decent support team, the operation was finally running smoothly—just as it all came to a close.



SEPTEMBER 2010

BOOK REVIEW MEL SCRIPTING FOR MAYA ANIMATORS
gamedeveloper
POSTMORTEM: NAMCO's PAC-MAN

DECEMBER 2005

PRODUCT REVIEW AUTODESK 3DS MAX 8 SOFTWARE
gamedeveloper
THE BUZZ ON HARMONIX'S GUITAR HERO

FEBRUARY 2006

MARCH 2010

DECEMBER 2007

JANUARY 2009

4. SLOW BLEED INTO SKELETON CREW

I joined the magazine in September 2004. Just a scant few months before that, there had been an editor-in-chief, a managing editor, a features editor, and an assistant editor. There was an on-site group of people to support magazine production, an in-house sales staff, and much more.

In the interest of retaining profitability for the magazine, the parent company took these positions and cut them or merged them into other departments. Most of the additional work got pushed on the editors. We became responsible for final magazine production, which there had previously been an entire department of five people for. We became the direct line to sales, instead of having an ad production person in between. We started doing our own layout.

Whereas before, we were just taking care of the words and occasionally helping with image sourcing, we were now in charge of almost every aspect of production. The increasing workload made us editors increasingly frazzled, and it became harder and harder to see the long view with the magazine. We were scrambling to get a single issue out. What's more, we couldn't afford to hire professionals much of the time, so we would hire amateurs or neophytes and train them. This put a lot of strain on the senior staff (read: me) as we got everyone up to speed.

When you consider the fact that none of our authors were professional writers (they're all game developers, many of whom don't claim English as a first language), a lot of these articles needed a lot of massaging, multiple rewrites, and hardcore copyedits. This became harder and harder to get right as we had to do more and more things that weren't editing.

Streamlining our production process eventually got this to a stable state, but it was never an easy job. In the end, we were able to make a good magazine with a skeleton crew, but just making a good magazine wasn't enough to keep up with the times.

5. WHO ARE YOU?

Who will mourn our passing? This is a strong admission here, but we never really, truly knew who our audience was. As a print publication, all we had to go on were the few emails we got, and our interactions with developers at trade shows. We never knew for sure if we were serving our audience, because our audience kept changing. Our surveys showed our audience included a lot of programmers, so we tried to accommodate them—but we also wanted to serve every aspect of the community. Case in point: Only two percent of surveyed readers called themselves audio professionals. And yet we maintained an audio column until the very end, because we believe audio is an absolutely vital part of game development.

We got a lot of flak from programmers for not having articles that were innovative enough. But we'd ask them what sorts of things they'd like to see, and they'd grow silent. "How about you write something you'd like to see," we'd ask—but nobody was falling over themselves to offer solutions, only criticisms. We had to try to predict what might be important, asking our advisers and peers for input. We certainly weren't perfect, and I'll be the first to admit there were some issues that were basically duds. But we had no real hard data to go on; your average Gamasutra article got more feedback than we would on an entire issue.

A number of people who said to me they would miss the magazine when they found out about its closure also admitted they got it for free and threw it away every month. Who was our audience, really? Will they miss us? Will you?

DAYS GONE BY

I will miss the magazine. All of us editors (and former editors) will. We did our best to try to help the industry we love, by providing a resource, and a venue for the various voices of our craft. We wanted to help you make better games, and I can only hope we had some small impact in our two decades of work. I honestly thought we might be the last game magazine left alive after all the others fell, because we were profitable, and couldn't slim down any further. But a small profit sometimes just isn't worth the time spent working on GDC and the company's other endeavors.

Thanks to all the editors, all the contributors, all the copyeditors, all the advisers, and all those who ever wrote to us, or spoke to us about the things we were doing. For me, at least, that was what kept me doing this. The fact that I knew someone out there, somewhere, was reading. Maybe, just maybe, they took something away from our words. f

***

Brandon Sheffield is director of Oakland, California–based Necrosoft Games, and editor emeritus of Game Developer *magazine. He has worked on over a dozen titles, and is currently developing two small-team games for PlayStation Mobile. Follow him on Twitter via @necrosofty.*

WE WANT THE BEST
MSTWNTED

TOP 30 DEVELOPERS OF ALL TIME

GAME DEVELOPER **AND GAMASUTRA STAFF PAY HOMAGE TO THE BEST DEVELOPMENT STUDIOS IN THE BUSINESS—EVER**

For a few years now, we at *Game Developer* have consulted with the Gamasutra editors to determine a list of the top 30 developers of the past year (from June to June, that is). Normally, this list is meant to recognize the studios that have shown excellence in creativity or in business, in product or production; in other words, the people out there doing work that inspires the rest of us by virtue of being new, better, or different.

When it comes to the last issue of *Game Developer*, however, a simple list of the last year's best simply won't do. Instead, we chose to assemble a list of the greatest game developers of all time. What follows is a list of 30 studios that have left (and in some cases, continue to leave) an indelible mark on the medium of video games for generations to come.

Note that whenever possible we've gone out of our way to avoid recognizing developers solely for being the first to do something in video games—our medium's pioneers are important, but we wouldn't have room left in the list for anyone else. So we've generally tried to stick to the last 30 years of game development or so, and focus on the studios that we think have shaped the current era.

NINTENDO ENTERTAINMENT ANALYSIS AND DEVELOPMENT

KYOTO, JAPAN

Odds are that most people reading this article owe their fascination with video games in some part to Nintendo EAD (SUPER MARIO BROS., LEGEND OF ZELDA), so including it on this list is a no-brainer. However, we feel compelled to point out that it's not just their work from almost 30 years ago that earned Nintendo EAD a spot; we're impressed by the way they consistently manage to push video games as a whole in new directions. Plus, as people who grew up with video games, we think there's something to be said for knowing there is someone out there keeping Nintendo's genuinely friendly blue-skies aesthetic alive.

LUCASFILM GAMES/LUCASARTS

SAN FRANCISCO, CALIFORNIA

We have a soft spot in our hearts for Lucasfilm Games (later LucasArts) and their adventure games: MONKEY ISLAND, MANIAC MANSION, FULL THROTTLE, and GRIM FANDANGO, among others. When it comes to their impact on the industry as a whole, though, we think Other Ocean chief creative officer Mike Mika said it best in the May 2013 issue of *Game Developer*: "Even today, my fantasy of what game development nirvana feels like stems from my experience playing those games, and the insinuation that they were created in the most liberating and creative environment on earth." Well said.

ID SOFTWARE

RICHARDSON, TEXAS

For better or worse, we can trace the dominance of the first-person shooter straight back to id Software and its seminal titles WOLFENSTEIN 3D, DOOM, and QUAKE. To be sure, these games have left an indelible impact on the game industry—particularly when it comes to 3D graphics programming and networked multiplayer, for example—but we're also inclined to honor them for their generally dev-friendly attitude toward game development, as demonstrated by their encouraging and open stance toward modding, and their habit of open-sourcing their id Tech engines. Also, we're keeping our fingers crossed for a gritty COMMANDER KEEN reboot.

EPIC GAMES

CARY, NORTH CAROLINA

Certainly, Epic's list of games is impressive enough—nobody can doubt the impact that UNREAL and GEARS OF WAR have left on the modern game industry. But we're including them on this list for the Unreal Engine itself. Between the raw power of Unreal Engine and UDK's relative ease of use, Epic has proven that they're not just good at making games, they're good at making tools to help people make games—and make those games look better than ever. (Just for the record, we'd accept a gritty JAZZ JACKRABBIT reboot, too.)

BUNGIE

BELLEVUE, WASHINGTON

We knew Bungie was cool way back in their Macintosh-only days, when we were devouring the terminal text in MARATHON and tossing grenades in MYTH: THE FALLEN LORDS, but we never could have imagined that Bungie would have basically carried Microsoft's Xbox and Xbox 360 with HALO. We're sure that lots has changed since the studio grew from three people to over 300, but we've remained impressed by how Bungie has consistently maintained a deeply thoughtful combination of technology, design, and creative direction over the years.

MAXIS

WALNUT CREEK/EMERYVILLE, CALIFORNIA

We can't write this list without acknowledging Maxis for SIMCITY and THE SIMS. During its prime, Maxis was eerily capable of making sim games just complex enough to be engaging and fun—and also managed to smash a few PC game sales records in the process.

MICROPROSE

ALAMEDA, CALIFORNIA

From early flight sims (F-15 STRIKE EAGLE, F-19 STEALTH FIGHTER) to Sid Meier's CIVILIZATION and PIRATES!, MicroProse's PC games legacy in both development and publishing throughout the '80s and '90s is impeccable. This legacy would be later left in Firaxis Games's hands, when studio co-founder Sid Meier left MicroProse with Jeff Briggs and Brian Reynolds to continue their strategy game direction with further CIVILIZATION titles (and more recently, XCOM: ENEMY UNKNOWN).

TREASURE

TOKYO, JAPAN

There's little we can say about Treasure that doesn't ultimately boil down to "They make really, really good games." Treasure makes great originals (IKARUGA, GUNSTAR HEROES), Treasure makes great licensed games (YU YU HAKUSHO, BLEACH: DARK SOULS), Treasure makes great shoot-'em-ups and beat-'em-ups and fighting games... The list goes on. We bow to Treasure's mastery of the craft.

SQUARE

TOKYO, JAPAN

Love it or hate it, FINAL FANTASY started as a last-ditch effort to stay in the game industry, grew into an international phenomenon, and ended up growing into a multimedia empire of its own. Chunsoft's DRAGON QUEST may have pioneered the modern Japanese role-playing game as we know it, but we credit FINAL FANTASY with bringing the genre to such prominence in the 1990s and early 2000s (also, CHRONO TRIGGER).

BIOWARE

EDMONTON, CANADA

Role-playing games stress a dev studio's ability to build worlds; BioWare excels at writing them. BALDUR'S GATE, NEVERWINTER NIGHTS, STAR WARS: KNIGHTS OF THE OLD REPUBLIC, DRAGON AGE, and MASS EFFECT are all representative of BioWare's unparalleled abilities to take worlds and games with complicated rule sets and histories (like FORGOTTEN REALMS and STAR WARS, for example) and distill them into the essence of video games as storytelling. Considering the game industry as a whole is often characterized as violence-obsessed and juvenile, we like that BioWare managed to build its reputation on good writing, simple design, and compelling worlds.

ORIGIN SYSTEMS

MANCHESTER, NEW HAMPSHIRE

Simply listing the WING COMMANDER or the ULTIMA series doesn't do Origin Systems justice. Origin's characteristic polish and technical capability inspired the next generation of developers—and many Origin devs went on to do great things (Paul Steed, Raph Koster, Warren Spector) as well. Like their motto said, they created worlds.

SEGA AM2

TOKYO, JAPAN

Sega in general is near and dear to our hearts, but we wanted to give Yu Suzuki's team a special nod for their string of great work from AFTER BURNER and OUTRUN to VIRTUA FIGHTER and SHENMUE. AM2 made arcades a magical place to be, and we miss that.

LOOKING GLASS STUDIOS

CAMBRIDGE, MASSACHUSETTS

You'd think that it'd take more than two notable game franchises to get yourself on *Game Developer's* list of the Top 30 Devs of All Time, but Looking Glass Studios earned this just by the strength of THIEF and SYSTEM SHOCK alone. (We know they did other games, but every time someone suggested including Looking Glass, it was just for those two.) Naturally, we can't bring up Looking Glass without mentioning that Ken Levine started his career there on THIEF before eventually going on to found Irrational Games and later creating BIOSHOCK.

GAME FREAK

TOKYO, JAPAN

It's easy to forget that POKÉMON wasn't always a multimedia empire (complete with yearly feature film releases!). It all started out with series creator and studio co-founder Satoshi Tajiri looking to design a game that replicated the childhood thrill of catching and collecting insects—and gradually evolved into arguably a blueprint for accessible/addictive game design and cross-media success. And it was super effective.

VALVE SOFTWARE

BELLEVUE, WASHINGTON

Certainly, Valve's track record when it comes to their actual games is impressive; the legacy they've left thus far with HALF-LIFE, PORTAL, TEAM FORTRESS 2, LEFT 4 DEAD, and COUNTER-STRIKE pretty much guaranteed them a spot on this list. We could mention Steam, with which Valve parlayed their success making games into arguably the most influential digital games marketplace. Really, though, we want to acknowledge them for, from all accounts, being a truly developer-led studio that understands the value of getting brilliant people together and treating them well—which is an example we think the industry as a whole ought to take to heart.

Sherman, Set the Wayback Machine to...

While we're acknowledging the top game devs of all time, we figured we might as well take a moment to recognize the folks who set the stage for the modern video game, too. Humans have been playing games far longer than we've had computers—and computer technology has a longer history than most of us tend to realize.

ADA LOVELACE

Ada Lovelace is widely considered history's first programmer. Born in 1815 to the poet Lord Byron and Anne Isabella Milbanke, Milbanke encouraged her daughter to study mathematics from a young age. In 1842 she was commissioned to translate Charles Babbage's paper on his Analytical Engine, an early mechanical general-purpose computer. In the course of expanding upon Babbage's findings, Lovelace developed what we now recognize as the world's first computer algorithm. Lovelace remains an inspiration for women in technology to this day.

ALAN TURING

Known as the father of artificial intelligence, Alan Turing began his professional career as a fellow at King's College at the unheard-of age of 22. In 1936 Turing outlined the concept of the "a-machine" (what we now know as a Turing machine), a hypothetical device capable of simulating any computer algorithm. His Turing test, by which machine intelligence can "fool" a human participant, remains a reference point in the field of AI to this day.

GRACE HOPPER

Grace Hopper enlisted with the United States Naval Reserve during World War II, where she served on the Navy's Mark I computer-programming staff. Following the war, Hopper stayed on with the Navy to work as a research fellow at Harvard, where she coined the term "debugging" (after a literal moth became stuck in a Mark II machine). In her post-Harvard work Hopper developed some of the world's first compiler programming languages, including FLOW-MATIC, a forerunner to COBOL.

IAN SOMMERVILLE

An electronics technician and computer programmer, Ian Sommerville came to prominence among the Beat Generation of writers and artists. In 1960 he programmed the random sequence generator used by Brion Gysin in his cut-up technique, a Dadaist literary style later introduced to, and popularized by, William S. Burroughs. A lover and "systems adviser" to Burroughs, Sommerville collaborated with the author to produce "Silver Smoke of Dreams" and also developed the Dreamachine, a stroboscope device billed as "the first object to be seen with the eyes closed." We think he'd have made for a great indie game dev.

GARY GYGAX

As co-creator of *Dungeons & Dragons*, Gary Gygax was instrumental in evolving the tabletop wargaming scene of the late 1960s and early 1970s, providing the foundation not just for later generations of rules-based role-playing and strategy games but also an expansive body of role-playing lore.

-Kris Ligman

BLIZZARD ENTERTAINMENT

IRVINE, CALIFORNIA

When Blizzard made WARCRAFT, it seemed like everyone wanted in on real-time strategy games. When Blizzard made WORLD OF WARCRAFT, everyone wanted to make an MMO. We don't know what's in the water at Blizzard HQ, but whatever it is has enabled them to take an existing genre and turn it into an industrywide trend.

CAPCOM

OSAKA, JAPAN

Whenever possible, we've tried to avoid recognizing a massive game company on this list in favor of highlighting specific dev teams. With Capcom, we simply can't do this, because it'd mean picking favorites between the folks who made MEGA MAN, STREET FIGHTER, and RESIDENT EVIL (just to name a few!). Fact is, Capcom knows how to make a darn good video game, and that has kept them in the center of the industry for decades.

KOJIMA PRODUCTIONS

TOKYO, JAPAN

To be clear, when we mention Kojima Productions, we're retroactively including the work that the core team did as part of Konami during the original METAL GEAR SOLID days. The METAL GEAR SOLID series embodies some of the most creative fusions between game design and storytelling we've ever seen in video games before. Yes, it gets weird sometimes, but we're glad that Hideo Kojima is out there making big-budget games that get a little weird.

BULLFROG PRODUCTIONS

GUILDFORD, ENGLAND

We can't talk about inspirational devs without bringing up Bullfrog Productions, the studio that brought us POPULOUS, SYNDICATE, THEME PARK, and, of course, co-founder Peter Molyneux, who would go on to found Lionhead Studios (FABLE, BLACK AND WHITE). We like that Molyneux chases after ways to tease out the medium's potential for emotional engagement—and in doing so, inspires other devs to do so as well.

MOJANG

STOCKHOLM, SWEDEN

Mojang makes MINECRAFT, Mojang sells MINECRAFT, Mojang makes millions and millions of dollars. Practically overnight, Mojang became a fairy tale to tell starving indies: Work hard on ideas you believe in, do right by your players, and the money will come. So Mojang is on this list because what they've done with MINECRAFT alone is an amazing accomplishment—and also because we, too, want to believe.

UBISOFT MONTREAL

MONTREAL, CANADA

When we were putting together this list, we found that it was easy enough to pick out studios that had landmark games early on in the industry's history, or studios that emerged as dominant players in the current era. Ubisoft Montreal nearly flew under our radar—not because they don't deserve a spot, but because it's strangely harder to recognize a great studio when they're still around and doing great work like SPLINTER CELL, FAR CRY, ASSASSIN'S CREED, and the rebooted PRINCE OF PERSIA series. We like to think that in 20 years (forever, in this biz), Ubisoft Montreal will still be there, quietly cranking out a handful of top triple-A titles year after year.

RIOT GAMES

CULVER CITY, CALIFORNIA

Riot Games basically took a WARCRAFT III custom map and turned it into an international phenomenon that spawned a genre of its own—the multiplayer online battle arena, or "MOBA." What's more, they did so by championing the then-unproven free-to-play business model (well, unproven in American markets, anyway), and architecting a studio culture built solely around making LEAGUE OF LEGENDS the best darn game they could make it. Since 2006, Riot has focused solely on developing and sustaining one game, which we think is a testament to how much they believe in their game. Here's to the next seven years.

PLATINUM GAMES/CLOVER STUDIOS

OSAKA, JAPAN

We're going to cheat and cover two studios in one entry, since Platinum Games's core trio of Shinji Mikami, Atsushi Inaba, and Hideki Kamiya previously worked together as Capcom subsidiary Clover Studios. Between the two studios, this dream team is responsible for VIEWTIFUL JOE, OKAMI, GOD HAND, BAYONETTA, and MADWORLD. We like that Platinum Games is unafraid to unleash their creative energies and take risks; we love that they do this and make games that just feel solid, satisfying, and lovingly crafted.

THATGAMECOMPANY

LOS ANGELES, CALIFORNIA

Even after FLOW and FLOWER, naysayers and traditionalists had no problem dismissing thatgamecompany's work. After JOURNEY'S runaway success in 2012, though, we don't think anyone can ignore thatgamecompany anymore. To us, the fact that thatgamecompany's emotion-directed development approach can thrive and flourish indicates that our industry and our medium are gradually maturing in their capacity for creative expression—which we think was a long time in coming.

ELECTRONIC ARTS - WE SEE FARTHER DEVS

N/A

Can a computer make you cry? That's the question posed in an early print advertisement for Electronic Arts in its early days during the mid-'80s. Nowadays, EA's reputation isn't quite so clean, but we couldn't put together a top 30 list without acknowledging their roots as a publisher for what was basically a loosely affiliated dev collective composed of: Bill Budge (PINBALL CONSTRUCTION SET), Anne Westfall and Jon Freeman (ARCHON, MURDER ON THE ZINDERNEUF), Danielle Bunten Berry (M.U.L.E., SEVEN CITIES OF GOLD), John Field (AXIS ASSASSIN), David Maynard (WORMS?), and Mike Abbot and Matt Alexander (HARD HAT MACK). It's somewhat comforting to know that video games have a long history of artists and iconoclasts looking to push the medium further.

SIERRA ON-LINE

OAKHURST, CALIFORNIA

Sierra On-Line (originally On-Line Systems) started from humble beginnings with MYSTERY HOUSE, the first-ever adventure game with graphics. With KING'S QUEST, SPACE QUEST, QUEST FOR GLORY, GABRIEL KNIGHT, and, yes, LEISURE SUIT LARRY, however, they grew the adventure game genre into an integral part of the video game landscape. Sierra On-Line may not have made it into the current generation of games, but their impact on the industry and subsequent generations of developers lives on.

RARE

TWYCROSS, ENGLAND

Rare started out making games for the NES at a ridiculous clip (releasing 18 games in 1990 alone!), went on to partner with Nintendo as a second-party developer, and then got purchased by Microsoft to make games for the Xbox and Xbox 360. When you think about it, it's kind of amazing that the same studio responsible for BATTLETOADS, DONKEY KONG COUNTRY, and KILLER INSTINCT also did GOLDENEYE 007 and KINECT SPORTS. That kind of longevity in a cutthroat industry like this is, well, rare.

MAME TEAM

N/A

As we work our way through this list of legendary game dev studios, it's worth taking a moment to honor the all-volunteer team behind MAME (Multiple Arcade Machine Emulator). Fact is, we're really glad there are people working behind the scenes to preserve a playable, accurate version of our arcade game history, especially when that kind of preservation work entails breaking old proprietary encryption schemes and reverse-engineering undocumented system architecture. Keep doing what you're doing, folks.

INFOCOM

CAMBRIDGE, MASSACHUSETTS

Sometimes it seems like games are a medium in search of a master storyteller to show us all how it's done—which is funny, because Infocom did exactly that with only text in 1980. Their works of interactive fiction—most notably the ZORK series, of course—demonstrated that games could engage players in ways other media couldn't. What's more, with later releases like A MIND FOREVER VOYAGING and LEATHER GODDESSES OF PHOBOS, they tackled mature themes meant for adult audiences that developers almost 30 years later still have problems sorting out.

HAL LABORATORY

TOKYO, JAPAN

Between their early work on the MSX to their more well-known NES and Game Boy games (ADVENTURES OF LOLO, KIRBY'S DREAM LAND), HAL Laboratory was probably Japan's best third-party development studio in the late '80s to the early '90s. We think that Nintendo buying HAL in 1992 was probably one of the best decisions they've ever made: Not only did HAL go on to make EARTHBOUND and SUPER SMASH BROS., they also hired a "genius programmer" straight out of college named Satoru Iwata, who would eventually go on to succeed Hiroshi Yamauchi as president of Nintendo.

Honorable @Mentions

***

Naughty Dog

"They are incapable of making a bad game."—@Luc1ferous

***

SNK

Modular arcade systems are pretty neat. We, too, miss the MVS.

***

Rockstar Games

GRAND THEFT AUTO is more or less a cultural icon.

***

Insomniac Games

Insomniac got a whole lot of love from our Twitter readership for making great games. We at *Game Developer* like Insomniac for a different reason: They're always willing to talk dev tech and technique, often within the pages of our magazine.

***

From Software

While From Software has been around forever and done a ton of awesome work, we suspect that most of the nominators are folks who really, really, really like DARK SOULS and DEMON'S SOULS.

***

Bethesda Softworks

"Bethesda has taken away probably 300 hours of my life."—@Randy_Floustine

***

Double Fine

Everyone loves Tim Schafer.

***

Black Isle Studios

Game devs really liked FALLOUT 2 and PLANESCAPE: TORMENT, apparently.

***

CCP Games

EVE ONLINE is the game everyone loves to read about, whether we're reading stories about the players doing amazing things in-game, or stories from the devs as they explain how they designed the game to facilitate those amazing player stories.

DIRTY GAME DEV TRICKS_Staff

ILLUSTRATION: JUAN RAMIREZ

DIRTY GAME DEV TRICKS

STORIES OF DEADLINE-DRIVEN TRICKS AND HACKS

One of *Game Developer*'s most popular features was our "Dirty Coding Tricks" bit from 2009, where we got devs to open up about some of the ugly hacks they've resorted to in order to make a ship deadline or pass certification. Well, we're back with nine new from-the-trenches stories, including a few unorthodox tricks from other dev disciplines besides programming. So read on, revel in your colleagues' ingenuity, and relax—because you're not the only one that pulls out a dirty trick under pressure.

CROUCHING RSX, HIDDEN TEXTURE ASSETS

Joe Valenzuela, Insomniac Games

This trick was on the PS3: We on the Insomniac engine team had some textures that we wanted distributed with our engine/tools release. These were things like noise textures and source input for full-screen filter effects. For some unimportant reasons, we didn't want to distribute these as actual asset files, so instead we converted them to binary arrays and compiled them into the executable. There was one downside, though—we wanted these to be in a different chunk of memory (RSX visible), so we would end up copying them out and just wasting the memory for the source.

The PS3 toolchain had a link feature to put certain sections in RSX memory, but it requires using 1MB pages, and in our case it would have wasted 700k. Instead, we added a new data section in the executable that aliased the bss (the "bss alias" or "balias" section). We had something like 3MB of bss in our final builds, so there was more than enough space to hide some texture assets. We ran some code as early as we could in the crt initialization to initialize the destination memory, copy the assets out, and then re-initialize the bss to 0.

Believe it or not, it worked! There was a little tweaking to accommodate hidden bss use and toolchain updates, but overall it was pretty straightforward.

THIS IS NOT THE BUG YOU'RE LOOKING FOR

Brett Douville, LucasArts

In early 2002, we were readying STAR WARS: JEDI STARFIGHTER for submission to Sony. One niggling TCR bug remained, which was that the controller's analog stick functionality would shut off while we were loading our post-mission cutscenes, causing the red light in the center of the controller to go off as well. This bug had shown up when we updated to a library version required by Sony, and the programmer who had originally written both the movie-loading code and the IOP logic for the controller itself had left LucasArts some months before.

In the hopes of narrowing down the problem quickly in code I didn't understand, I inserted seven screen clears in different colors in the code I determined to be the likely source of the problem, hoping that I could at least narrow it down to one or two sections of code by checking what color the screen was when the analog controls turned off. But when I then tried to reproduce the bug, it had disappeared.

It's an old programming adage that if you don't understand the cause, you can't be said to have fixed the bug. But in this case, we were two or three days away from our intended submission date and missing it would have been a big deal. So I changed all the screen clears to black, marked the bug fixed, and called it a day. We shipped on time, with no TCR showstoppers.

(S)ELF-EXPLOITATION

Jonathan Garrett, Insomniac Games

RATCHET AND CLANK: UP YOUR ARSENAL was an online title which shipped without the ability to patch either code or data. Which was unfortunate.

The game downloads and displays an End User License Agreement each time it's launched. This is an ascii string stored in a static buffer. This buffer is filled from the server without checking that the size is within the buffer's capacity.

We exploited this fact to cause the EULA download to overflow the static buffer far enough to also overwrite a known global variable. This variable happened to be the function callback handler for a specific network packet. Once this handler was installed, we could send the network packet to cause a jump to the address in the overwritten global. The address was a pointer to some payload code which was stored earlier in the EULA data.

Valuable data existed between the real end of the EULA buffer and the overwritten global, so the first job of the payload code was to restore this trashed data. Once that was done things were back to normal and the actual patching work could be done.

One complication is that the EULA text is copied with strcpy. And strcpy ends when it finds a 0 byte (which is usually the end of the string). Our string contained code which often contains 0 bytes. So we mutated the compiled code such that it contained no zero bytes and had a carefully crafted piece of bootstrap asm to un-mutate it.

By the end, the hack looked like this:

1. Send oversized EULA
2. Overflow EULA buffer, miscellaneous data, callback handler pointer
3. Send packet to trigger handler
4. Game jumps to bootstrap code pointed to by handler
5. Bootstrap decodes payload data
6. Payload downloads and restores stomped miscellaneous data
7. Patch executes

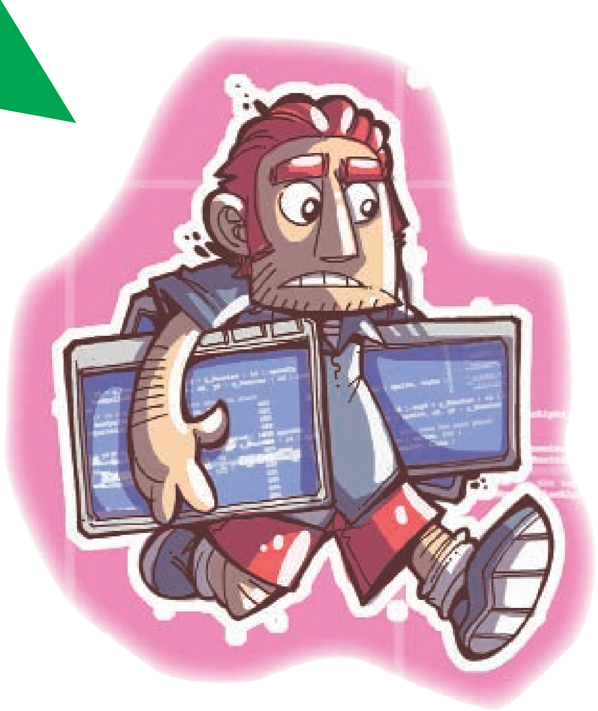
Takeaways: Include patching code in your shipped game, and don't use unbounded strcpy.

JAMMING THE CARTRIDGE

Michael A. Carr-Robb-John, Monolith Productions

In 1993 I was finishing off DESERT STRIKE; I was doing the conversion from the 16-bit Mega Drive / Genesis to its humble little brother, the 8-bit Master System. The game was exceeding the desired cartridge size by about 12K, and going to the next size cartridge was out of the question. Today, 12K sounds incredibly small, but back then it was a major deal. During development, I had budgeted and planned all the sound and graphic resources and they were within their limits. The only section I hadn't been so strict on was the code. In those days, games were written in assembly language—in this specific case, Z80 assembly—so I had only one option left. I spent a week going through and finding redundant code and rewriting things to use a smaller memory footprint (usually at the cost of being more processor intensive).

MAKING GAMES IS NOT A HOBBY



By the time I had finished, the game fit onto a cartridge with just 98 bytes left! The game was burned onto ROM and tested for a few days by the chaps in QA before being submitted to Sega for certification. Unfortunately it didn't pass on the first time, and the required fixes pretty quickly used up those 98 bytes. I think when we did publish, there were only 6 bytes free!

THE DALTON ALLOCATOR

Jonathan Adamczewski, Insomniac Games

Toward the end of one project, we discovered that after playing through the game for many hours, movies would not trigger when they were supposed to. Fragmentation within one of our memory allocation heaps had made it impossible to reliably allocate the large blocks of memory that were needed for full-screen movie playback.

We needed to find a way to be sure that we could always get the memory we needed. However, it was too late in the project and too risky to consider defragmentation of the heap, we didn't have enough spare memory available to set aside exclusively for movie playback, and taking space away from other systems at that stage was impractical.

It was clear that when movies were playing, there were many other systems sitting idle, so we gave some consideration to "borrowing" memory from them. However, these other potential sources were either too small, or else also suffered from the same kind of fragmentation problems that we were already seeing. There was plenty of free space available for the GPU, but for various reasons we could not use that for the movie-playback buffers. So we needed another source of space.

While looking into another problem, a particular pattern of memory allocations within our main game-asset heap stood out: When the game started up, a number of assets were loaded from disk, and once in memory were never modified. Some of those assets were very large. None of them were needed while a movie was playing.

This gave us an idea: What if we were to copy the contents of one of the larger asset files to somewhere else in memory? The GPU's memory would work just fine as a place to temporarily stash the asset, and then we could temporarily reuse the space in the asset heap for movie playback, copying the asset data back from GPU memory when the movie finished.

So that's what we did. We chose the largest set of animation clips used for one of the game's heroes (named Dalton). The animation system was switched off, animation clips copied into the GPU's memory space, and the memory handed to the movie-playback system. At the end of the movie, the animation data was copied back to where it belonged, the animation system restarted, and the game proceeded as if nothing horrible had happened. The implementation ended up being fairly straightforward, though the process does span a number of frames to ensure that all the systems involved are safely synchronized with each other through each step of the process.

The memory for the movie system has since been described as coming from "the Dalton Allocator" after the character whose animation clip memory was usurped.

(I've since discovered that there's some history of this kind of thing at the studio—of both stashing random things in GPU memory, and failing to set aside enough space earlier in the project...)

CERTIFICATION HEADACHE

Michael Carr-Robb-John, Monolith Productions

Anyone that has ever written a game on a console knows about the certification headache. For the most part, certification is simply common sense and good practices, but there are one or two "requirements" that just seem to be nothing more than a way to cause developers problems. One such requirement that we've had to deal with is that from the time the user chooses to run your game, you *must* be displaying the first presentation screen within four seconds. If you have a large executable, it can take at least two or three seconds just for it to load before you get control, and you still have to load a whole bunch of visuals and sounds in order to present the main menu.

My game was taking 26 seconds from the time the user selected the game to the time it displayed the first presentation screen, so I could already feel that headache starting. My first job was to isolate what was required to display the menus, and put off loading specific global data until it was actually needed. Surprisingly, this had a bigger impact than I was expecting; it knocked off over nine seconds. Many tweaks later, I had managed to knock it down to around 10 seconds, but I just couldn't get it to load any faster. While discussing the problem with another engineer (best problem-solving method I have ever found), the solution dawned on us.

The specific console also has another requirement that two specific screens are displayed by your game before progressing into the menu system. What is also of interest is that the user *must* be prevented from skipping the screens for a set amount of time. Let's see now—if we loaded only those two screens, and did most of the menu loading while the player is watching those screens...*voila*—5.5 seconds to load and display.

Unfortunately, that still was not enough to satisfy the letter of the requirement, so we eventually had to ask for a waiver, which was granted (probably because we were so close).

PAINTING SOUNDS

Edward J. Douglas, Flying Helmet Games

I was running cinematics on a long-running racing series. Our scenes were a mix of straightforward startgrids and fancier action sequences. As we iterated on further sequels, our ambitions with the scenes got greater and greater, but our technology iterated at a slower pace.

You see, the cars would be animated by a QA "stunt driver" to get the base motion, then hacked up in a 3D animation program to adjust the timing and placement of the action, and re-exported to our in-game cinematic tool where playback would simulate all the physics and engine behavior. Lots of gameplay data was captured, including gas and brake information from the controller and represented in meta-data in the 3D file, then reproduced in-game. The idea was that this would drive the car audio system as well.

The problem came a few sequels down, where our scenes were very complicated, with a mix of hand-animated car action and recorded capture. The old tricks of using the engine meta-data to drive simple audio rev samples for our cars wouldn't work anymore—the data just wasn't there! The audio team couldn't post-process the audio like a movie, because any scene could have any car in it, depending on player's choice and modifications, so the audio needed to be procedural. But by this time, our games had won numerous awards for audio, especially for the car engine sounds, so we were determined to make it work.

Coming on to beta, things were looking dicey, but a combination of ingenuity and madness between our cinematics, audio, and AI engineering teams found the solution. The gas and brake metadata was represented by a float scale on a cube in the 3D scene. If an artist went in and "drew" curves in a keyframe editor like 3DS Max, they could draw in the car engine sounds they wanted. A few members of the audio team rushed to learn 3DS Max, and by using their intuition of how rev patterns should look, they drew in the animation, exported all the scenes, and squeezed it all in in time. It sounded great, but after this last-minute hack, we knew we'd need something more robust if we'd continue with the same tech-base for the next sequel.

...Or so I thought. I left the studio after that game, and a few years later I met an audio guy who had joined that team after I left. It wasn't long before I realized they never upgraded the tech, and he was the "engine rev painter" guy for the latest sequel.

...AND ONE FOR GOOD LUCK

Richard Morwood

I had a list of background textures, and I coded the game to show each one as they scrolled across the screen. One of my background images was being skipped and I couldn't figure out why, even after spending a lot of time debugging. Deadline was five days away, so I just stuck an extra reference to the texture in the background list. Ta-da! No more "skipped" background. :)

HR HACKS

Ben Burbank

When I used to work for a very big company, one of the employees figured out that the best way to advance his career was to write negative performance reviews for as many co-workers as possible. This resulted in him receiving a higher annual staff ranking, which in turn led to larger bonuses and stock grants. It eventually becomes difficult, he told me, because you need to make sure to only review people with different managers, so nobody can catch on to your ruse. My trick for avoiding this cycle was to quit and go work someplace much smaller and awesome.

HONORABLE MENTION: NICE SAVE

[Editor's note: This isn't, strictly speaking, a dirty game dev trick—but we figured it's a handy way to use job skills for real-world problems. Also, it's a sweet story.]

Chris Pruett, Robot Invader

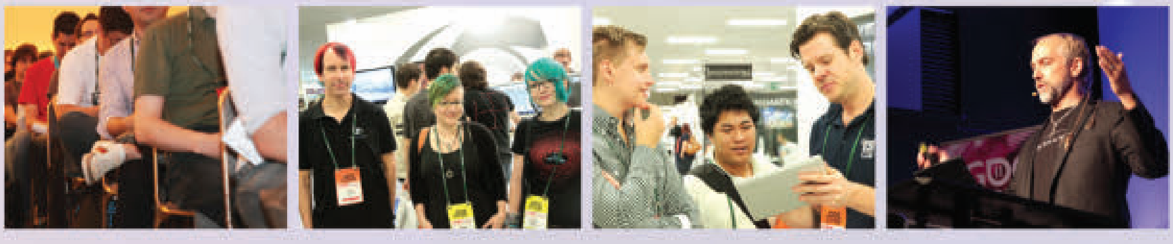
My wife doesn't play a lot of video games, but one series she's been hooked on since childhood is DRAGON QUEST. A few years ago, she started playing DRAGON QUEST 7 on my aging PlayStation. After putting about 80 hours into it (which, as I understand it, is about three-quarters of the way through the main quest), she discovered, to her horror, that her save file had become corrupted. It appeared in the continue menu, but was grayed-out and could not be selected. She was devastated. She was angry. She swore never to play games again.

I found a used DexDrive, a device for reading and writing PS1 memory cards with a PC, on eBay for $15. I didn't tell my wife that I was trying to fix her save—I didn't want to get her hopes up and I didn't actually think it would be possible. Presumably the data was irrecoverably damaged, a lost cause. On the other hand, I thought, it couldn't hurt to try.

With the DexDrive, I was able to dump the broken save to a PC and examine it in a hex editor. I ended up printing it out and marking the hex up with a highlighter; though PS1 saves came in 8kb blocks, printing 8kb as 16-columns of hexadecimal data comes out to a lot of pages. Working with a unofficial spec written by the author of a PS1 emulator, I located the main chucks of the data: the header, the icon image, and lastly, the save data itself. Unfortunately, decoding the raw game state data proved challenging; after a few days I decided that it was going to require a lot more work than I had planned.

Instead, I concentrated on the header portion of the data. Because of the placement of the icon (which is at a consistent offset from the top of the file and easy to identify as pixel data in hex), I could tell exactly where the header started and stopped. If the continue menu could tell that the save was busted, maybe it was just the header that was broken. I tested this theory by copying the header data from some other save I got off the Internet, and pasting it over the header section of my wife's broken save. Then I saved this data back to the memory card and loaded it up.

Miraculously, it worked. The continue menu showed stats from the other save, but once loaded her game was completely restored. Between ordering the DexDrive and patching the save, the whole process had taken me about three weeks. That night I booted the game up and showed her the continue menu with the strangely named save. She loaded it up and was very surprised to find her progress, her characters, her stats, and her items all as she left them. She was pretty excited, but before we could talk about it she was off to complete another dungeon.



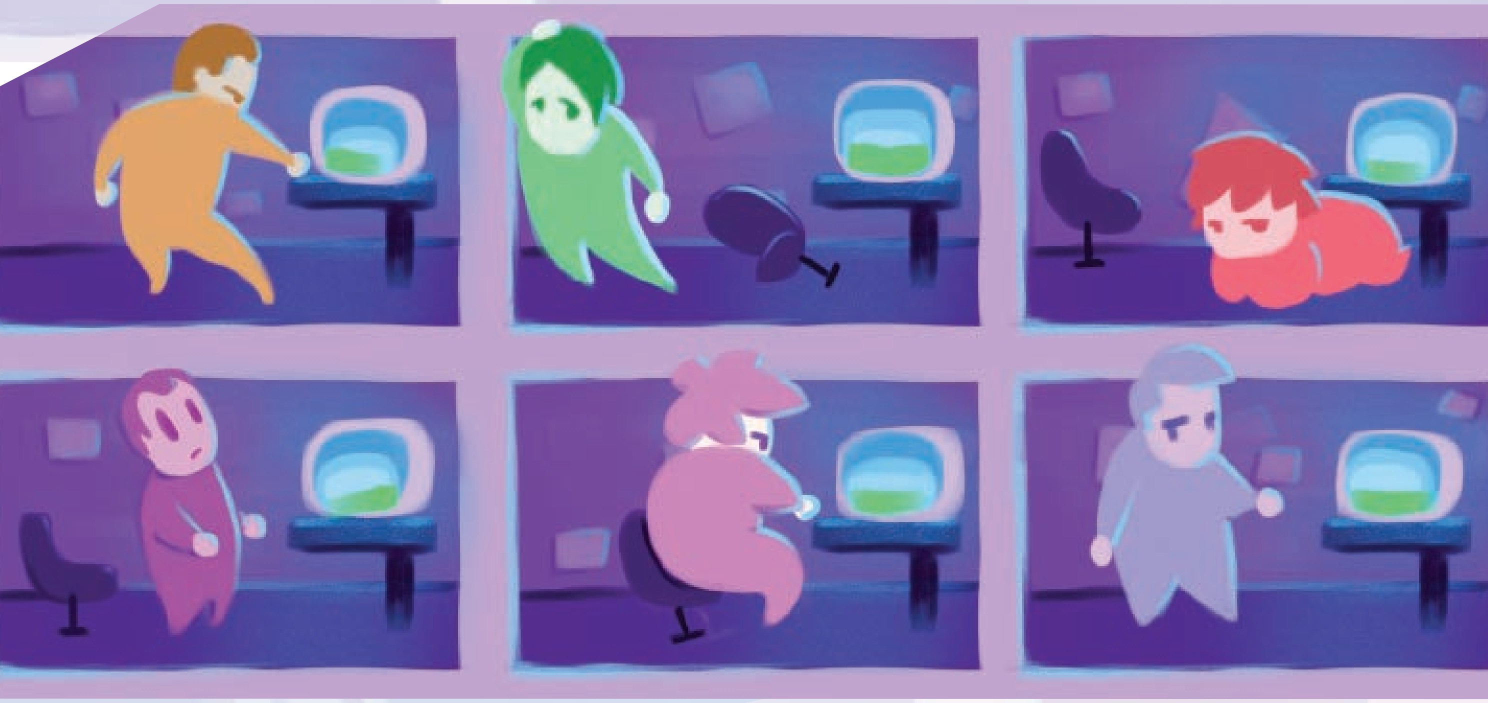
GAME OVER

GAME DEVELOPER'S SURVIVAL KIT FOR THE FUTURE OF VIDEO GAMES

BY STAFF

For 19 years now, *Game Developer* has borne witness to the game industry's comings and goings. As a magazine, we have endeavored to provide a space for game creators to swap tips and techniques, speak frankly about the challenges of making games, highlight and recognize good work and new talent, offer big-picture analysis of industrywide trends, and advocate for more compassionate development practices, all in the name of helping people make better games.

But *Game Developer* is over. We know that all our readers will go on to do amazing things in video games, but we won't be able to make a magazine that will help you out. So we've decided to pool the *Game Developer* and Gamasutra brain trusts and lay out what we see as the game industry's key changes, challenges, and opportunities coming in the future—because you'll have to tackle them without *Game Developer*'s help.



TRIPLE-A, INDIES, AND GAME DEV AS A FULL-TIME JOB

We know we aren't the first ones to say that we think the writing is on the wall when it comes to triple-A game development. Practically everything new and interesting to happen in game development over the last few years has ended up taking a bite out of triple-A's pie: Indies, better off-the-shelf dev tools, the rise of mobile and social games (and their associated app stores), and the emergence of free-to-play online games all thrive in areas triple-A dev cannot.

Meanwhile, budgets for producing and marketing triple-A games continue to increase even though their respective profit margins appear to be growing narrower. We don't see triple-A disappearing entirely, but we *do* think it will be relegated to a less central role—perhaps one akin to the blockbuster movie in film, with only a few major releases per year from the few industry giants capable of funding and sustaining that kind of multiyear development effort.

Instead of making bigger games by hiring more dev teams, we expect that the industry will be forced to develop more efficient tools that will enable smaller teams to take advantage of ever more powerful hardware (whether that hardware is in a PC, a dedicated console, a mobile device, or something else). Toward that end, we expect more and more devs to come to rely on third-party tools and middleware to make games look better and play smarter without relying on massive dev teams (and massive dev budgets).

Over the last few years, we've seen game developers generally focus less on solving cutting-edge tech problems and more on pushing the design envelope, creating a unique visual motif, or solving business problems (monetization design and user acquisition, for example); we expect this trend to continue, and for off-the-shelf dev tools to become increasingly more sophisticated and easy to use.

As game dev tools continue to evolve, more and more people will be able to make games—games that take risks (both of the business kind and the creative kind) that the industry giants won't be able to. We think that the incredible rush of creative energy coming from indie developers will continue—and that their innovation in creativity and business will expand the overall market for games by attracting new audiences and finding new ways to convince existing audiences to pay for games.

On the other hand, this might not bode well for the full-time game developer job market. Perhaps game development will fall into similar patterns as writing, photography, film, and music: something that is really easy for individuals or small groups to do on their own (and possibly even make money from), but harder to break into the professional class than it already is now. More people will make games in their spare time, just like people start blogs or garage bands, while doing something else to pay the bills.

Studios looking to stay ahead of this change will have to do a better job building their brand, defining a stronger "personality" in their games, and attracting an audience that sticks around from one game to another (in other words, developing an audience that is attached to the developer itself, not just the IP of the games it makes)—all while keeping their dev processes as efficient and lean as possible *and* hanging on to talented teams. It won't be easy.

LEAVING "VIDEO GAMES" BEHIND

Think of the term "video game" as analogous to the term "motion picture"; both phrases describe a medium in the simplest, most literal sense possible. Motion pictures are pictures that are moving, and video games are games that you play with a video display of some sort. But when taken literally, "motion picture" could describe movies, television, commercials, music videos, or anything else that happens to contain video content that we watch.

When it comes down to it, the term "motion picture" is simply too broad and vague for us to actually use in everyday conversation. Even though all of the above types of "motion picture" are usually created with the same basic tools (a camera), and there is overlap in the skills necessary to produce each different kind of "motion picture," we typically consider the different formats of motion pictures different media entirely, with different artistic techniques, delivery mechanisms, consumption patterns, and so on.

This is where "video games" are headed, too. "Video games" is quickly becoming a catchall term for all kinds of media that have very little in common with each other besides the fact that they exist in a virtual space and are authored with a set of similar tools. Some games are virtual toys and play sets; some are sports; some are virtual community spaces; some are interactive narrative experiences; and so on.

We expect virtual, interactive entertainment to become the *de facto* method of popular communication—integrating itself alongside music, film, and other traditionally passive forms of art rather than in opposition to it—and each "genre" of game will eventually grow into a medium unto itself. Imagine having a *Game Developer* or GDC specifically devoted to developing competitive-eSport games, or toy games, or story games, and you have the idea.

So, how can game developers develop their skills *now* to prepare for the future game industry? We suggest that you're probably best served by focusing on the ins and outs of a specific sector of games; when it comes to the skills you need to make a fantastic MMO, or a competition-focused sport-game, or a heart-wrenching episodic drama, we suspect that they will only become more specialized and less cross-applicable as the industry matures. We wouldn't expect Steven Spielberg to be good at inventing football, after all, so expecting game developers to be similarly multitalented seems like a losing proposition to us.

(The irony of people working in print publishing offering advice on future-proofing skills—in the last issue of the magazine, no less—is not lost on us, by the way.)

NEW VOICES FOR VIDEO GAMES

Recently, we've seen conversations about inclusion, diversity, and the game industry pop up at trade shows and conferences, on web sites, forums, Twitter, and just about everywhere else. This is not a new conversation, though it is perhaps louder now than it has been in recent memory.

Each year, our Salary Survey pegs the gender ratio in the game industry at about 89% male, give or take a percentage point or two. For comparison's sake, a 2011 report by the U.S. Department of Commerce called "Women in STEM: A Gender Gap to Innovation" found that women held 24% of STEM jobs (science, technology, engineering, and mathematics), and 27% of jobs specifically categorized as "computer science and math."

ILLUSTRATIONS: BRENT PORTER

OVER

You read that right: The game industry's gender ratio is *twice* as bad as the overall STEM fields' ratio.

This is a problem. There is no legitimate normative reason why creating video games should be overwhelmingly a function performed by men. Fortunately, we're beginning to see the barriers to creating, distributing, and playing games come crumbling down, which has given rise to quite a few new groups of people making and playing games. What's more, these new voices in video games are often making games for themselves and each other, which serves to expand the medium's potential both from a creative aspect (discovering new messages and mechanics) and a business aspect (popularizing video games as an entertainment form to new consumer demographics, and deepening games' reach for a higher yearly per-person spend). It's good for everyone, and it's good for games.

The barrier to entry isn't technical; it's cultural. We take it as a basic truth that people get into this business in order to make games that they themselves would like to play. When the industry is historically composed of young men making games for other young men to play, you end up creating a culture around the medium that is also by men, for men. And, at its worst, this culture can be insular, defensive, exclusionary, and downright nasty when prodded to change its ways. Thus far, games have done an excellent job of making money—as an industry, we've eclipsed both recorded music and Hollywood—but as a medium of mass communication it still isn't taken very seriously. As long as game development is primarily the domain of young men, we don't see this changing significantly.

We've framed this conversation so far strictly in terms of gender, but the same could be said for sexuality, race, economic class, and so forth. It's no coincidence, we think, that criticism of game industry's same-ness, particularly in the triple-A mainstream, has continued to grow louder as we've seen more not-white, not-male, not-straight, not-middle-class people start to make games. And when we look at the devs that are admired within the industry—the people who do the creative work that inspires us to do better—we're seeing that more of these folks are the not-white, not-male, not-straight, not-middle-class people who are gradually making games their medium, too.

As a trend, we expect this to continue in fits and spurts, and we're looking forward to that. However, it would be negligent on our part to assume that this trend will continue without asking that good people out there continue to do their hard work to make the game development community more supportive and welcoming. (Many of these people are contributors and friends of *Game Developer*, so if you're reading this: Thank you.)

We all owe it to ourselves, our colleagues, and our community to make video games as accessible and open as possible, however we can. This could mean initiating and encouraging institutional changes and ideological shifts to further break down these walls; or, perhaps, we can just start by scrutinizing our own individual behavior and attitudes and systematically eliminating the ones which may cause ourselves or our colleagues to behave like assholes despite our best intentions.

COMMUNITY MANAGEMENT IS IMPORTANT

As journalists, we understand journalism. That doesn't just mean that we understand how to write stories. It may seem like a simple job, isolated from reality, but in 2013, it sure isn't. Just like you have a big picture of your industry and your career, so must we. This is the last issue of *Game Developer* magazine, so this might sound especially portentous, but look, you have a choice here, too. Not only is it increasingly obvious that you have the opportunity to take control of your relationship with your players, but it has also become quite clear that the players prefer it that way.

The truth of the matter is that we expected things to be much further along this road by now. Why is Nintendo the only major platform holder that completely controls its game announcements, going straight to fans with its Nintendo Direct presentations? Many of the big developers have community strategies—usually hiring from their player bases or recruiting ex-journalists—but these strategies look a little myopic at times, inasmuch as they seem based around preserving the status quo of community rather than expanding its role either outward or inward. You can slap a pearlescent purple coat of paint on a 1990s QUAKE clan, but that's what it is, at its heart. The rest is marketing, and that's not a real connection.

This work doesn't all have to be done by the big guys, and it doesn't all have to be done in one specific way. People are now waking up to the idea of crowdfunding—fine. But your community is a huge asset across all vectors, and you need smart people figuring out how to best harness it, not just communicate to it or manipulate it for short-term gain. You should be thinking very specifically about how your community likes to interact with you, what they like about your game ,who they are, and how to reach them. We've seen developers use community members for bug tracking, design ideas—whatever. What's more, this is particularly important for smaller devs who might not have direct access to their player communities (if you're publishing mostly on mobile app stores, for example)—track them down and forge a strong relationship with them. It's an investment in your future (and your future games). At the same time, when the community comes running with the pitchforks, defend your creative vision; if you don't respect it, no one else will.

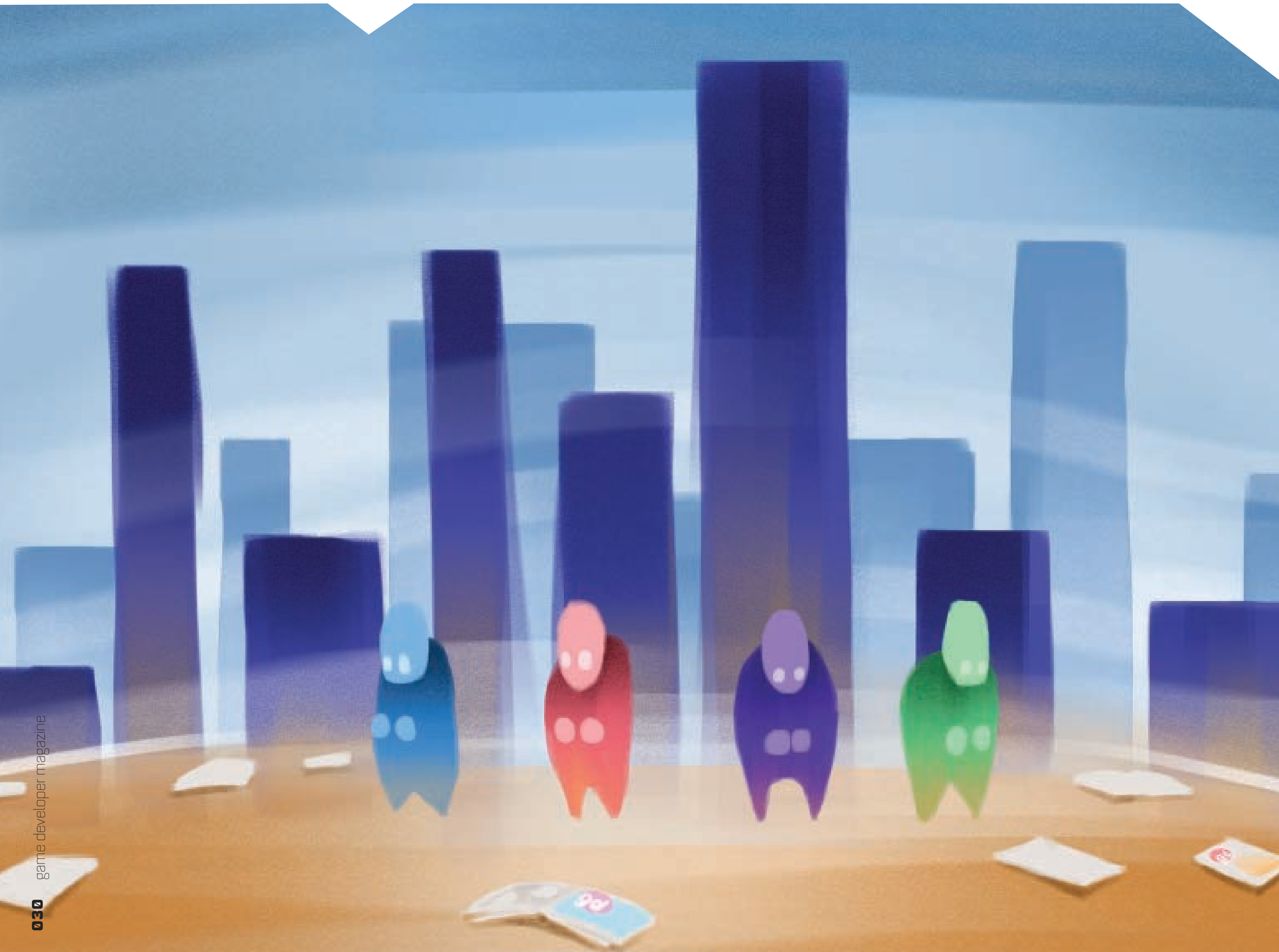
CRUNCH, BURNOUT, LAYOFFS

The game industry is subsidized largely by the enthusiasm and passion of its employees. At least, "passion" is the only reason we can imagine that devs would enter an industry where layoffs are routine, unpaid overtime is the norm rather than the exception, and job applicants need multiple shipped titles and years of experience under their belts even for entry-level positions.

For some devs, working conditions have gotten better since the "EA Spouse" days. But crunch is still seen as a relatively normal part of a standard game development cycle—one that is still sometimes worn as a perverse badge of pride—and we think this is unsustainable and wrong.

Here's the deal: If you can't afford to make a game without overworking your employees, *you can't afford to make it*. Make it cheaper. Find a way to use a prototype or minimum viable build to bring in more funding. Make a different game. Budgeting for software dev projects is hard, but once your projects *routinely* rely on unpaid overtime to ship, you can't use that as an excuse. You can probably get away with it, thanks to a yearly crop of fresh game program grads, but it's a lousy thing to do.

Game developers, by and large, are smart, hardworking people. Smart, hardworking people eventually figure out that other industries are willing to treat them better. When you're young and hungry, you might be willing to put up with the bullshit for The Love Of The Game, but at some point you will probably sit down and think that it simply isn't worth it any more. The human cost of game development can be measured in friendships lost and family time missed by every person in the list of credits at the end of a game. We don't think it's worth it. And sooner or later,



we find that many devs tend to agree, especially once they're looking to settle down, start families, buy houses, and so on.

What's more, the endless cycle of crunch, burnout, and layoffs holds the industry back from a quality standpoint. When you let dev talent leave your studio office, you're losing all the experience and expertise they've cultivated specifically for *building your games*. When we let dev talent leave the industry, every amazing game they could have made walks out the door with them.

A strong work ethic is a fine thing, and worth being proud of. But we don't think it's a good thing to value your work ethic over other important things, like friends and family. And we especially don't think people should be proud of a culture of overwork, especially when that culture has deleterious effects on not only your health and your relationships, but your colleagues' as well. (Also, we suspect that the people who stand to profit from your overwork do not have your best interests in mind.)

MAKE QA BETTER

Many smart people have spoken about how one can judge another's character by observing how they treat others; we think the most quotable version is from J.K. Rowling's book *Harry Potter and the Goblet of Fire: "If you want to know what a man's like, take a good look at how he treats his inferiors, not his equals."* This quote is from a powerful wizard discussing how another powerful wizard treats his servant elf, but it's not that far from describing QA's relationship to the rest of the game industry.

Fact is, in most of the games industry, QA gets no respect. It's a career dead end, the pay is awful, and the best thing that happens to most QA folks is that they get routed into a different discipline. In any other industry, this is called an "internship"; game dev has created an entire caste of people meant to do menial work. We admit that we're just as guilty as the rest of the industry, in that regard; we don't publish a regular QA column like we do for other dev disciplines, and we rarely address it elsewhere in the magazine.

The thing is, it doesn't have to be this way. If QA was seen as a field worth staying in for its own sake—and compensated as such—we could easily see studios develop a competitive edge by building vastly improved testing methodologies and incorporating them in different stages of the development process, so as to make sure the right feedback and testing reaches the right people at the right time. Career QA specialists could go on to develop specialties related to different aspects of game development. Imagine having a dedicated QA veteran working in tandem with an artist or audio designer to more efficiently ferret out graphical glitches or audio malfunctions, or a game design QA specialist devoted to homing in on balance issues, and you have the idea. Considering more and more publishers are tying bonuses to Metacritic performance (which doesn't allow for changed review scores and thus is heavily affected by bugs and release-day issues), we don't think it's impossible that properly investing in QA would have a significant boost on a studio's bottom line.

Beyond the money stuff, though, we think there's a real human cost to making QA a slog—especially when it's the *de facto* point of entry for the games industry if you aren't already a whiz programmer or artist. We can appreciate that every profession demands a certain amount of dues-paying in the beginning (we too were interns once), but from some of the stories we've heard, QA seems like a cold, capitalist version of fraternity hazing.

Think of it this way: QA is the entry point for the industry. As a discipline, QA is largely characterized by endless drudge work for low pay, and a lot of hopping around from contract gig to contract gig until you can find a studio that likes you enough to take you on as a

QA lead or entry-level in a different discipline. Logically speaking, it follows that the people who made it into the industry through QA have already established that they're willing to work long hours of drudge work for low money and minimal job security. Now look at the labor issues that extend across the entire industry, not just QA—long hours with relatively low hourly pay, and alarmingly frequent layoffs. We don't think this is a coincidence.

CONSOLES ARE LOSING THE CONSOLE WAR

If that sentiment annoys you, then maybe you'll be okay. Maybe you're annoyed because you know you're in a studio that's on top of the console food chain, with all of the premier talent, the biggest franchise, the biggest marketing resources. So someone saying "consoles are doomed" to you is akin to someone telling you your bike is doomed—while you're riding it.

"Your bike is doomed, my friend."
"No it's not, look. I'm a good cyclist, I'm riding it right now, and it's perfectly fine, asshole."

You may be fine, as long as Microsoft, Sony, and Nintendo don't screw things up too much for you. Except your success on game consoles ultimately relies on whether these companies can move hardware units and fight off competition from emerging platforms, and the latest hardware launches surely are not instilling confidence in us about the long-term viability of the dedicated video game console.

Ask yourself this: Can Microsoft, Sony and Nintendo be trusted to move hardware units? The old guard of traditional hardware makers is already answering the question. Nintendo's answering with an emphatic "no" right now with the Wii U, which is driving operating losses so hard that Nintendo missed their forecasts like a small-town weatherman. Even the 3DS, which course-corrected slowly after its launch with a price drop (and which many assume is doing just fine), is behind Nintendo's own expectations. This, in turn, is causing traditional publishers like EA and Activision to feel gun-shy. Nintendo's not making the best argument for the future of the console business.

There's another young console, albeit one that's handheld, that might also serve as a microcosm for the declining state of consoles—the PlayStation Vita. It has some things going for it, not the least of which is its nice hardware, and a network tied to PlayStation Plus's well-played digital business model. But this thing is tanking, and the small installed base is just not giving developers a good reason to make a game for it, because it is priced at a level that forces it to compete against both the Nintendo 3DS *and* smartphones.

As for future home consoles on the horizon, like the PlayStation 4: Sony seems to have done well so far with the initial details: It has plenty of fast RAM, an x86 processor, PC architecture, and seemingly strong relations with big and small developers, even right now before the console has launched. But as dev-friendly as it might be, we think that new consoles will have to compete against increasingly TV-friendly PCs, whose core audiences will be playing Steam games on one front, and on the other front will be Nintendo and Microsoft, fighting over this shrinking piece of console pie. So can we trust Sony to move plenty of PlayStation 4s? The company expects dollar sales of its game unit to increase in the next fiscal year thanks to PS4's launch. For now the answer is "maybe," in the near-term anyway, when early adopters spend their cash (and mainstreamers play their iPads instead). And barring a drastic change from Microsoft, we'll presume that they're in the same "maybe" boat.

In order for consoles to stand a chance, they'll need to compete with each other's platform on value, openness, power, and convenience. The next console must be powerful enough to offer game experiences that clearly separate console games from mobile games (and compete with PCs); accessible enough to devs that indies and small studios aren't turned away from including those consoles on their new, cool stuff; convenient enough to convince players to turn to their TVs instead of their PCs (or their smartphones, for that matter); all while competing in price against Steam sales and 99-cent app store price points. We don't think that this is an impossible task (see: PlayStation Plus and Sony's recent overtures toward indie devs, for example) but it looks like an uphill battle *already*, and given technology's lightning-quick pace these days, we don't think it'll get any easier over the next few years.

On the bright side, even though the business of consoles (as we know them) will certainly have a difficult future and will play a role in lots of headaches and heartaches for our readers, developers should take note right now that it's not all bad news. People will still want to pay for the "console experience," even if that experience won't be on a console platform. We already see this starting to happen in the mobile market, and that initiative is growing fast with more "mid-core" developers rising up (just wait till the market is ready for "full-core"!). If you're not already, take into consideration what you need to do to diversify your business and your skills so you can survive: You are your own life raft. Consoles may be doomed, but your career may still be bright.

STEPPING DOWN FROM THE SOAPBOX

So, that's where we stand—consider this feature to be a few years' worth of Game Plan editorials. Make no mistake: We believe in this industry, in its future, and its immense creative power. We just won't be able to use these pages to help it along anymore. f

God of War: Ascension

034

POSTMORTEM_June/July 2013

BY WHITNEY WADE
AND CHACKO SONNY

GOD OF WAR: ASCENSION is the fourth mainline installment in the award-winning GOD OF WAR series. Development started almost immediately after the completion of GOD OF WAR III, and we set out to exceed the achievements of the prior game—deliver something bigger, better, and more awe-inspiring while adding something new to the franchise. Ultimately, the product succeeded on many fronts, establishing a strong multiplayer game within the franchise, but fell slightly short of the near-universal critical acclaim for GOD OF WAR III.

Over the course of development, we faced a number of unique challenges, but we managed to take advantage of the Santa Monica Studios team's key strengths to complete the project on time, within budget, and to our quality standards. This postmortem reflects some of the things that went right, some of the things that went wrong, and the key lessons we learned over the course of building this game.

WHAT WENT **RIGHT**

MULTIPLAYER BROUGHT NEW LIFE TO THE FRANCHISE The decision to add multiplayer to the traditionally single-player GOD OF WAR was controversial from the start, both with fans and team members. This departure from our core expertise forced us to reconsider how we made games, and in some ways, it enabled us to return to the principles that helped create the original GOD OF WAR: extensive playtesting, brutal feedback, constant iteration. This invigorated our technical and creative forces. The MP team rekindled our "underdog" spirit—we needed to prove to critics, players, and ourselves that we could deliver the entirety of the GOD OF WAR experience in multiplayer. In light of the recent spate of "tacked on" (critics' words, not ours) MP additions to other games, this was an extremely difficult task, as many dismissed the MP game outright before even trying it. People on the team have never been more passionate about proving something, and after our MP Beta, we had converted legions of fans who had either been on the fence or outright opposed to it into our strongest advocates.

NEW HIRES BROUGHT NEW LIFE TO THE TEAM In the past, Sony Santa Monica has grown from within; many of our senior leaders are team members who have been here from the very beginning of the team. Over time, a number of team members who have left briefly for stints elsewhere in the industry have returned to SMS, acknowledging that there is something unique about the team, process, and products we create.

For ASCENSION, however, the scope of the game and the complexity of the entire project required that we grow the team dramatically. We acknowledged early on that we required a variety of new skills (multiplayer engineering and design, for example), and leadership expertise in order to successfully deliver the project we had defined. Through an aggressive hiring push, working in conjunction with on-site talent acquisition staff who helped execute our rigorous interview process, we were able to find key contributors in all disciplines and at all levels of seniority.

Our newer team members bring with them a fresh outlook and offer experience or ideas on how we can improve things. This new life is something that we don't take for granted, and we had to work hard to make sure that all team members still felt like they could bring their contributions to the table, even as we grew. This is something that went very right on this project—careful hiring pays off in the long term, and has led to one of the strongest Santa Monica Studio internal dev teams yet.

FRANCHISE STRENGTHS PROVIDED A SOLID FOUNDATION FOR INNOVATION AND QUALITY ASCENSION'S visual and gameplay spectacles, like the Hecatonchires at the beginning of the game, and the Fury Monster at the end of the game, would not exist without sequences from prior games like Cronos, Gaia, or Poseidon. For the Hecatonchires sequence, the team sought to exceed the complexity of the introductory sequence from GOD OF WAR III, which was a massive challenge. By building on the strengths of the prior games and leveraging the team's knowledge of how to create these sequences, we were able to focus on continually pushing the quality bar.

Critics have universally hailed the visuals in the game as the peak of the PS3 generation. Further, the gameplay innovations that were added—including new navigation mechanics, or the blade magics, for example—were the direct result of a desire to improve on systems from prior games.

STRONG MARKETING AND PR PARTNERSHIPS LED TO GLOBAL STRATEGY AND EXECUTION For this project, we had outstanding relationships with marketing and PR that started early and stayed consistent through the entire project. The teams were raring to go with ideas on how to push the boundaries of what we'd done before, but they also respected the amount of work we had on our plates to make a game that would live up to what we set out to accomplish. We revamped our visual campaign to tie into the key game art motif, but we also looked to completely rework godofwar.com, and make sure that our website became a destination for information and interaction more than it had ever been before.

While supporting these things took time from our senior leads, we worked together early on to define an overall plan that we wanted to support, instead of feeling like we were flying by the seat of our pants to support last-minute requests. Last-minute requests are normal for a lot of the marketing and PR process, of course, but we were able to reduce them by having a plan and communicating it outward to all territories, and getting global buy-in and team support. Having a clear master plan, as aggressive as ours was, went smoother than on any other project. Our marketing and PR teams are clear partners in the success of our project, having taken great care in supporting the franchise and the development team.

EARLY MP TESTING FORCED US TO THINK OF THE GAME AS AN EVOLVING SERVICE Multiplayer beta testing provided a chance to validate that the system and game design for the MP game would hold up to the scrutiny of a dedicated fanbase, and to the volume of players we'd see once the game launched. Since multiplayer was new to most of us, we wanted to make sure that we were buttoned up in the wild before launching. While we have certainly discovered a few issues post-launch, they have been manageable, and people can still play while we address them.

We released our game in March 2013, but we started beta testing in fall 2012. The beta testing proceeded as expected, revealing successive layers of issues that we would address, deploy fixes for, and then monitor in a live environment. The most important change from this testing was not the improvement to the software, but a fundamental change in how we planned for supporting the game once it went live. We had designed the team's workflow to enable them to quickly iterate on fixes, deploy them through infrastructure (testing, certification, patching), and coordinate with internal technology partners within Sony. We also developed the right systems to monitor, log, and interpret any changes we implemented. Also, the live testing environment gave us insight into how we should prioritize for key features, and drove our development tasking as we closed in on release.

CROSS-FUNCTIONAL LEADERSHIP ALLOWED US TO "FINISH STRONG" By the end of the game, we needed to focus and get it all done. There was still a lot of work to be done, and we needed to change up how we had approached completion on prior games. In the past we had created small pods to work on very specific areas of the game, like Cronos, for example.

We determined that each level needed key design and art leads to gather the massive volume of fixes completed in a day and make sure the levels stayed functional as we were putting the finishing touches in. The builds were locked at that point, as well, so we needed to make sure that not only the levels themselves stayed stable, but that the game as a whole didn't break. For example, we had a key artist and a key designer gather changes on behalf of the respective teams. This included lighting for art and cameras for design. Art would gather and playtest the entire level by lunch, and then design would gather and playtest by dinner.

At the end of the day, we had a completed level with that day's fixes. Not only did this help keep everything together and running, but it also helped with communication in the respective disciplines, as well as between art and design. It also created individual ownership, as well as responsibility to not let down others who were working in the same areas. This was one of those things that I slapped my forehead over and wished we'd put into place way earlier on the project.

WHAT WENT **WRONG**

SHORTENED PRE-PRODUCTION DUE TO SCHEDULE LIMITATIONS As one project ends, it's important to begin planning for the next. We hold to a very high standard during our finalizing process, and the reality is that GOD OF WAR takes all focus and all hands on deck to finish. Everything is custom, and every part requires a level of pride and perfection that only we put on ourselves, and it all comes together in the end.

Balancing that kind of singular focus against pre-production for the next project was a very big challenge. At the end of GOD OF WAR III, we were also doing some R&D experiments with an entirely new project. Our excitement ultimately came back around to telling a new GOD OF WAR story, but that conclusion came later than we would have liked, and we lost out on invaluable pre-production time. Between "all hands on deck" to finish the previous game, and experiments with new directions we didn't take, pre-production for ASCENSION was not in the place where Todd Papy (game director) and senior leadership wanted it to be when we began. This proved to be very difficult to recover from, and it did have a detrimental impact on the entirety of the project.

TECHNICAL DEBT DELAYED DELIVERY OF KEY MECHANICS While many of the systems that were improved for GOD OF WAR: ASCENSION built upon successful systems from prior games, some features, not least of which was multiplayer, required substantial rewrites that took longer than expected, and we couldn't deliver those mechanics until well after key deadlines. Our entire player navigation code was something that we had to completely change to allow for online play, which was initially assumed to be co-op and therefore had to work alongside the single-player campaign. That was a much bigger undertaking, and it didn't go as smoothly as we had hoped, so we lost of lot of mechanic development time in that effort. Ultimately, the single-player and multiplayer parts of the game were more delineated than was initially expected, and a less disruptive approach would, in hindsight, have been better.

While the wall-navigation system was ultimately smoother than in prior games, revising this system required additional effort from animation, environment art, design, and engineering, and by the time the system was fully implemented, we didn't have enough time to polish it sufficiently. In the end, we learned the hard way that we should have put more time into these elements up front, and better understood the scope and effort needed to finish them.

THE ROAD TO MULTIPLAYER WAS LONGER THAN EXPECTED We knew that we wanted to tackle multiplayer, but we didn't know what that meant in terms of creative and technical resources. We had never attempted multiplayer, and we didn't have a lot of team members who had that experience, so we created our own rules—some of which were successful, and many of which would have been identified as unsuccessful, earlier, if this had been an established multiplayer gameplay genre.

When we started thinking about multiplayer, we left the constraints wide open. We knew that we wanted to do more than tack on a bunch of playable Kratos characters in a competitive arena. We knew we wanted to do more than add a co-op bot, and most of all, we knew that whatever we did had to be awesome and GOD OF WAR in its epicness. What we didn't know was that every idea would feel too small upon the start. What we didn't know was how much work it would take to make the game we envisioned, and how many people would be required to successfully realize that vision. On top of that, we all didn't agree on what the game should be.

So we ventured on and did a lot of prototyping. We started down a co-op path and realized that it just wasn't going to be what we wanted. It wasn't until just before our first showing to the press and public that we really found the heart of what would become our final multiplayer game, and we tweaked it right up until Alpha. In the end, however, we're really happy with what we shipped, and we continue to hone the experience through patches and DLC.

SHARED LEADERSHIP FOR SINGLE- AND MULTIPLAYER TEAMS SLOWED PROGRESS As anyone who's built one knows, building a multiplayer game along with single-player is like making two separate games. The amount of attention required for mode-specific issues requires dedicated leadership for each game type. While we had the majority of the team fully dedicated to either multiplayer or single-player, some key leaders, including Todd and Chacko, were split between single-player and multiplayer.

As we were getting ready for our big press announce, which exclusively featured a multiplayer reveal, our entire focus was on getting multiplayer ready for that. During that time, the single-player game got very little senior leadership attention. By the time we refocused onto single-player and toward our big single-player E3 debut, we ended up playing catch up, and then multiplayer suffered.

We juggled this way through the entire project, which also resulted in slower decision-making than desired on most everything. Luckily, our team was able to keep the various balls rolling, but we can imagine it was very frustrating for them to have that pressure and lack of leadership from the lead producer and game director.

STRUGGLING TO BALANCE STORY AND THE SERIES TRADITION OF "EPIC MOMENTS" We really wanted to change the way we told the story, and try to go deeper emotionally with Kratos's storyline, giving the players more context and emotional substance to experience. We had a lot of initial ideas and even prototyped visual cues that we played with at the start of the game. We felt good about the direction. The story and script as a whole met that criteria, too.

Finding the right balance of story and the series' staple "Epic Moments" provided an unexpectedly complex challenge. As important as it was for us that we find new ways to tell the story—and the kind of story we were trying to tell—it was always critical that we also find ways to top the jaw-dropping setpieces of previous games. As we closed in on finishing the game, we decided to retain our primary focus on the game's biggest moments—those "epic" moments and setpieces. Unfortunately, this came at the cost of narrowing our story-telling vision. We're proud of so many moments in ASCENSION, but the finished game did miss many of the storytelling ambitions we had hoped to deliver to both ourselves and our audience.

GOD OF WAR: CONCLUSION In the end, GOD OF WAR: ASCENSION became a strong addition to the series, a chance for Santa Monica Studio to grow (both as individual devs and collectively as a team, in terms of their multiplayer experience), and also opened new doors for the franchise with a vibrant, active multiplayer community.

Our key takeaways over the course of the project were a better functional understanding of our deeply integrated production process, and the true scope of a dedicated multiplayer experience from concept to online implementation. All in all, ASCENSION required a tremendous amount of coordination and communication between team members in different disciplines. Ultimately, we found the solution to our challenges lies with better, more robust planning and prioritization up front. The team and the senior leadership learned a great deal on ASCENSION, and even now we're incorporating the key lessons into our process on our new projects. pm

***

Whitney Wade and Chacko Sonny are the senior producer and director, respectively, for internal development at Sony Santa Monica.

MONOPRICE GRAPHICS DRAWING TABLET

Graphics tablets have become ubiquitous because, simply put, they make drawing, painting, or editing with a computer feel natural. Tablets are an alternative to the mouse, and to some degree the keyboard, and as such, they may relieve the repetitive stresses of working with a computer. No CG studio is complete without one.

So which tablet should you purchase? In this review, we'll take a look at Monoprice's budget-friendly Graphic Drawing Tablet (MP1060-HA60) as a choice for new artists looking to dip their toes into computer illustration. Note that the Monoprice Graphic Drawing Tablet does not have a specific name, so for the purposes of this review, we'll simply call the MP1060-HA60 tablet "the Monoprice."

MEETING THE MONOPRICE

The drawing area of the Monoprice is 10 x 6.25 inches, which makes it a medium-sized tablet—a popular size range. The tablet's relevant specs weigh in as follows: 1024 levels of pressure sensitivity, 4000 lines-per-inch (LPI) resolution, and a refresh rate of 200 pictures per second (PPS). There is no tilt sensitivity, and the pen stylus has no eraser.

Pressure sensitivity specifies how much pressure the tablet can pick up. Typically, pressure sensitivity comes into play when drawing variable-width lines or strokes, working with opacity and color, and now when sculpting digitally. In general, professional-grade tablets start at 2048 levels. Resolution, measured in lines per inch (lpi), measures the amount of data the tablet can send to the computer, and the refresh rate is how fast the tablet updates the data sent to the computer. With HD displays becoming the norm, higher-resolution and faster tablets are better.

Thus, at 1024 levels of pressure sensitivity, the Monoprice is equivalent to the consumer line of Wacom Bamboo tablets or a Wacom Intuos 3 (the current release is Intuos 5). However, the lack of a pen eraser or tilt sensitivity groups the Monoprice with the more basic Bamboo tablets like the Connect ($80) or Splash ($80) instead of the high-end Bamboo Create. When it comes to specs and features, the Monoprice is a midrange consumer-level tablet that comes in a bit cheaper ($47) than its competitors.

Graphic Drawing Tablet
(MP1060-HA6)
www.monoprice.com

PRICE:
$47

SYSTEM REQUIREMENTS:
Windows 7, Vista or XP (SP3, 32- or 64-bit versions), Mac OS 10.4.0 (or later)

PROS:
1 Inexpensive
2 Simple to use
3 Generally works well

CONS:
1 Low resolution and low pressure-sensitivity levels
2 Cheap construction
3 Stylus requires AAA battery

TESTING THE TABLET

Of course, this doesn't mean it's a bad tablet. When I tested it with Adobe Photoshop, the Monoprice performed very well, and only a trained eye would notice any difference between the Monoprice's performance and that of higher-end tablets. (Think of it this way: Years ago, professional-quality tablets were limited to 1024 levels of pressure sensitivity, and that didn't stop computer artists from producing excellent work.)

In Adobe Illustrator, the tablet also worked well with the Paintbrush tool to produce nice variable-width strokes. However, because Illustrator produces vector strokes, it is easier to see the difference that higher pressure sensitivity makes, meaning the Monoprice's shortcomings were a bit more apparent.

While the Monoprice generally performs well, it has a few clear drawbacks. For starters, it does not support multiple displays, and while you can work around this limitation, it's still a major problem. Most of the bundled software that ships with the Monoprice does not work in Windows 7 or 8, and it's pretty gimmicky anyway. The tablet drivers don't work if there are other tablet drivers installed.

Also, the Monoprice pen stylus requires a single AAA battery, and it feels bulky and unbalanced. The reason the pen requires a battery is that unlike Wacom tablets, which use a passive system to detect user inputs, the Monoprice uses an active system in which the pen sends signals to the tablet. However, the Monoprice *does* work with after-market pens, so it may be possible to find a better stylus.

BUILD QUALITY The general construction of the Monoprice leaves much to be desired. Overall, the Monoprice feels flimsy, with a cheap plastic upper case and a thin metal bottom. The pen stylus also feels poorly constructed, and if you don't handle it properly while swapping out batteries it can be easily and permanently damaged. Even the penholder does not fit the pen very well, which again gives one an uneasy feeling about the entire unit. In stark contrast, Wacom tablets are sturdy and well constructed, and Wacom pens are tough and balanced to take years of use and abuse.

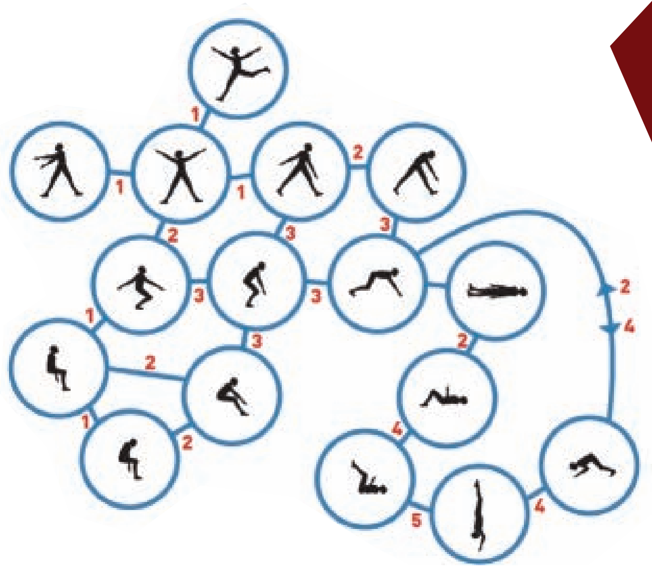
As mentioned before, Monoprice is the distributor of this tablet, not the manufacturer. In fact, the manufacturer is UC-Logic, an Asian firm that both sells tablets under its own name and produces tablet components for different brands. Unfortunately, it appears as though UC-Logic tablet technology has not changed in the better part of a decade, as older tablets using its components are nearly identical to the Monoprice in appearance and performance. It is a safe bet that if UC-Logic does not update its technology, tablets using its components will be outdated soon.

A SOLID STARTER-LEVEL TABLET Whether the Monoprice is a good pick for your studio depends largely on what you'd like to do with it. If you are a professional illustrator/designer/editor who depends on a tablet for everyday work, stick to professional-quality tablets. While it is *possible* to produce good work with the Monoprice, its technology is just too antiquated to take advantage of the features in modern displays and workstations, and I don't think its construction would hold up to everyday use. The Wacom line of Intuos tablets is much better suited for professional work, especially on high-end workstations.

However, the Monoprice is well suited for the amateur or hobbyist computer graphics artist, and its bargain price of $47 targets that audience perfectly. It's also not a bad pick for non-artists who need a cheap tablet for occasional simple graphics tasks.

***

MIKE DE LA FLOR is a freelance medical illustrator, instructor, and writer. He's the co-author of the recent title Digital Sculpting with Mudbox: Essential Tools and Techniques for Artists *as well as other CG titles.*



INNER PRODUCT, REVIEWED

A TECH TALK RETROSPECTIVE

Game programming has always been a discipline that embraces rapid change. Very few people code the same kinds of systems in the same ways they coded 15 years ago; fewer still expect to code the same kinds of systems in the same ways they're coding now in 15 years' time. And the pace of change is ever increasing. Radical change is around every corner. That being said, we should take care not to throw out all the hard-earned lessons of the past and take time to examine how they are still applicable and relevant today.

For this final installment of Inner Product, I have chosen a few of these lessons from back issues of *Game Developer* to share and discuss with you. Not everyone will agree with my choices or my opinions on those choices, of course—but if I only said things that everyone agreed with, I wouldn't really be saying anything at all! I invite you to draw your own conclusions.

"The ultimate weapon of game programmers is context. You know the needs of your game and can tailor the memory manager accordingly."
Brian Hixon, Daniel Martin, Rob Moore, Greg Schaefer, Richard Wifall, "Play by Play: Effective Memory Management," February 2002

There are very few problems that cannot be better solved by having better context. It remains a common programmer error to attempt to make solutions too generic, in order to try to solve all problems instead of simply the one at hand. In doing so, nine times out of 10, a programmer will add new constraints to the problem that will just make it worse.

"The obvious next step is to take this technique and put it in your game today. I want to be clear about this: I want better water in games, better lava, better liquid in general, and I want it now. All joking aside, though, I don't want to imply that you could actually drop this into your engine in an hour and a half; there are admittedly some obstacles with this technique."
Brian Sharp, "Moving Fluid: The Conclusion of a Two-part Series about Implicit Surfaces," August 2000

Ten years ago, Brian Sharp called for better fluids and better special effects. Today that call remains. While there have been some improvements and a few clever implementations, we still don't devote nearly enough time or resources to giving these effects a proper treatment in most games. Partially that's due to the problem posed in the previous quote—we try to find over-general solutions to problems that deserve a specific solution. But mostly, it's due to the same reason it was due to 10 years ago: It doesn't make enough difference to the final game.

"If all of DOOM was written in assembler and the programmer could manage the overhead correctly, Carmack theorizes it would only make the game 15% faster. And, although the main raycasting trace in Wolfenstein was written in assembly language, Carmack says he could write Wolfenstein faster in C because of today's better algorithm technology."
Alexander Antoniades, "Monsters from the Id: The Making of DOOM," 1994

Notwithstanding the nonsense phrase "algorithm technology," I would suggest this claim would be largely unchanged today. Compilers and optimization approaches have not improved all that significantly. There remains a significant percentage of cases where it's very easy to write an assembly routine that is much more performant than its C or C++ equivalent, mostly due to knowledge about the data (context) which the compiler cannot know, or ABI constraints.

That said, one thing has changed: The gap between instruction performance and memory performance is much, much wider. The biggest optimization opportunities are in proper management of data. Instruction-level performance is still important, and will remain so for some time. A great deal of code is still written in assembly, and that will remain the case. However, in a world where the abstraction tax is readily paid through scripting languages and Javascript UIs, the bar for improving performance is much lower now than it's ever been.

"Do not call a separate pixel-plotting routine in image-drawing code. Use inline code or functions instead. Think about how many calls it takes to draw one screen or image."
Matt Pritchard, "Ten Techniques for Faster Image Drawing," February 1995

Where there is one, there are many. That's more true today than it's ever been. And it's always been true. However, it's also true that more than ever before (given the spectacularly disastrous success of object-oriented programming), the actual data models and transformation costs are ignored by programmers, and dozens or even hundreds of routines are assigned to individual "objects" that all "act" independently and conspire to sabotage performance, maintainability, testability, and good sense.

"For frequently drawn images, use specific routines with hard-coded values instead of general-purpose routines. You can choose the variables you encode as constants and remove functionality that you don't need."
Matt Pritchard, "Ten Techniques for Faster Image Drawing," February 1995

Another reminder that "generic" is not a goal. The work of programmers is to transform data from one form to another. Sometimes, the same tool can be used many times to do that. Other times, it's much, much better to solve the more specific problem with a more specific solution. The goal is to solve the problem well. No player is going to give you kudos because you re-used a class in an inappropriately generic way.

"If you don't like math, well, computer graphics is math for the most part, so I'm not sure what to tell you. My goal is to describe the math in an accessible way, but I'm not going to hide the fact that math underlies everything about computer graphics, especially three-dimensional computer graphics."
Chris Hecker, "Perspective Texture Mapping, Part II: Rasterization," June/July 1995

Math remains at the heart of what we do. It's as important today as it was in 1995 to understand at least the fundamentals. But more than that, we have learned that you simply cannot hide it. Not just from programmers, but from all game developers. Artists cannot write good shaders without understanding at least the fundamentals of algebra, trigonometry, and geometry. And the efforts to hide that from them have more often than not ended in poor results—if not tears.

Designers cannot design good systems without understanding statistics and algebra at the very least, and those that do not often end up with very poor systems. Instead of trying to hide the mathy bits, it's usually better to teach it instead.

"The processor cache is usually an object of great fear, wonder, and misunderstanding. A friend of mine named Terje Mathisen says, 'All programming can be thought of as an exercise in caching.' Although Terje isn't talking specifically about the processor cache, this is a rule to live by when you're trying to optimize on modern processors. If we apply this idea to the processor cache and memory bandwidth, it means, 'Figure out how to put your important data in the cache and keep it there.' This may seem obvious, but keeping your data in the cache is more difficult than you might think."
Chris Hecker, "Memory Miscellanea," October/November 1995

The importance of hardware caches and understanding how to best utilize them has never been more important. [Reads redundantly.] There is simply no good excuse not to deeply understand whatever caching and data transfer hardware may exist on your devices, and it's poor practice to not design your data and systems around that knowledge as best you can.

The most likely bottleneck in any game is the accumulated time spent waiting for data. This bottleneck is usually spread so deeply through the fundamental design choices and abstractions in the systems that it cannot be fixed late in production without a rewrite, which late in production is impossible to justify. Instead, good stuff that would be valuable to the play experience ends up getting cut.

"The solution is easy. Keep two heaps. One heap is used to satisfy static allocation requests, and one for cached requests. You can simulate this with one heap by allocating the static requests at the bottom of the heap (growing up) and the cached requests at the top of the heap (growing down)."
Jonathan Clark, "Object Cache Management," February/March 1996

Over the last 20 years, a lot of effort has been put into garbage collection, dynamic memory management, and other types of "managed memory" systems. All this effort is based on one totally erroneous assumption: that managing memory is hard. Managing memory is no harder than managing any other kind of finite resource. It's the same as managing time within the frame, or scheduling out a project.

The solution is easy, and it has always been easy. Analyze what kind of data you have and what kinds of patterns of access and allocation are required, and divide it up into categories of similar things. Accessed together? Stored together. Lots of small items? Combine them. Unpredictable lifetimes? Keep that separate from the stuff that has predictable lifetimes. Many of the managed memory cures that have come into recent popularity are a great deal worse than the disease of simply managing your memory ever has been.

"A helpful debug tool is to zero out all freed memory. Then, whenever a reference to memory that is supposed to be free is used, a crash is likely to result and you can track it down easily."
Jonathan Clark, "Object Cache Management," February/March 1996

Another tip: Fill memory with a recognizable sentinel. "0xdeadbeef" is a popular choice. And since many allocations are aligned and have some unused area at the end, fill that too. That way, if you ever do accidentally stomp that area, you can test for it.

"One set of issues is with the workflow in creating procedural textures: It can be difficult to find the right person for the job, since building texture procedurally is effectively its own discipline."
Sean Barrett, "Hybrid Procedural Textures," October 2004

While most of us still rely very heavily on precreated textures, they are mixed very heavily in shaders. The world of hybrid procedural textures has come to pass, and it is very evident it has become its own discipline. That said, I'd really like to see more "procedure" in game textures. We could do a lot more to indicate things for the player—provide clues and signals and real-time feedback—if we weren't so generally concerned with trying to make things look "real"—a brick on

Figure 2. Pentium Cache Addressing

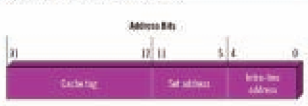
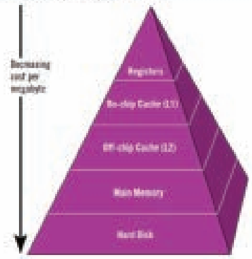


Figure 1. Component Cost vs. Amount



a real building isn't likely to change while you're looking at it very much. But there's no good reason why a "brick" in a game needs the same constraint.

"Bugs in software generally arise from complexity in control flow and complexity of state. A design that requires more 'if' statements is likely to have more bugs from incorrect conditionals; designs with more variables offer more opportunities to fail to maintain invariants between them. Programs with little flow control or state, e.g. scripts, offer few chances for bugs."
Sean Barrett, "Opening Doors," September 2004

Another often-ignored advantage to understanding what your source data is and what you need to transform it into (and simply doing that) is that it usually simplifies state checks and control flow by at least an order of magnitude. The more code has to work out what data is, and what it's supposed to become at runtime, the more chances you've introduced a bug.

"Gathering data solves only half the problem; to be useful, the data must be viewed by a human."
Sean Barrett, "Interactive Profiling Revisited," August 2004

Probably the most significant advancement of the last 20 years in games has been the realization of the importance of usability. Not just in the game interface (although those have gotten much better too), but in all aspects of development. That APIs are just interfaces for programmers and should be usability tested. That the usability of game creation tools has a direct impact on game quality. And that all the data in the world isn't much use if you don't know what specific action should be taken after you've obtained it.

"Ideally, we would like all character motions to be generated dynamically at run time: If AI-controlled characters want to sit down, they figure out how to move their muscles to get over to the chairs and place their butts firmly upon them. This is a very difficult thing to do, as a general problem it is far out of our reach."
Jonathan Blow, "Experiments I'd Like to Work On," June/July 2004

The argument among totally dynamic, motion capture, hand-animated, and hybrid animation models hasn't changed much. But I'm certain of one thing: The ideal is not that everything is generated dynamically at run time. That doesn't matter one bit. The ideal is that characters animate as the player expects them to animate. That the characters behave as the players expect them to behave. Sometimes very realistically, sometimes not at all. The ideal is that you can meet (or dare I hope, one day, exceed) the players' expectations for believability given the constraints of cost and resources, no matter how you do it.

"If you read non-sequentially from the disc, the data rate will drop to zero whenever the laser moves to a new location. In designing your game, you must somehow work around this problem. It's possible to cover up the seeks by playing previously buffered sound and video, but you could also display a static screen (text, for example) to distract the player."
Dan Teven and Vincent Lee, "Optimizing CD-ROM Performance under DOS/4GW," August/September 1996

While the problems of optical media have haunted us through CD, DVD, and Blu-ray, and we may hope to see the end of it one day, the larger problem will never change. You cannot design a system as complex as a game from end to end without understanding the constraints of how your data is transferred. From the source, whether that's optical media, HTTP server, or hard drive, to the various data caches, register files, and DMA engines. To design code first and not data first is to ignore the fundamental problem of programming real games on real hardware.

"The main point here is that you cannot expect the compiler to do much work for you beyond a rote translation of the code you write into native machine code. (With the incredible code generation bugs I've found, you sometimes can't even expect this.)"
Chris Hecker, "More Compiler Results, and What to Do About It," August/September 1996

The compiler is a tool. A programmer's work is not to write code. A programmer's work is to solve problems. Those problems are solved by transforming data. That data is transformed by code. The compiler is just a convenient tool in that chain. The programmer, ultimately, is responsible for making sure the problem is actually solved, and not understanding the tools and limitations of those tools will make that job much harder and the programmer less effective.

"The best-established algorithm for the general searching of optimal paths is A* (pronounced "A-star"). This heuristic search ranks each node by an estimate of the best route that goes through that node."
W. Bryan Stout, "Smart Moves: Intelligent Path-Finding," October/November 1996

In the intervening time, I wish we would have spent less time worrying about the best paths and a lot more time worrying about good, believable paths.

"It seems that we are dizzily cloning the clones of old clones. Wouldn't it be better in the long run to take the time to design something original once in a while?"
Chris Crawford, "Weeds and Oaks," June 1997

I dare to say with the popularity of game jams and the indie scene, more original games were made in the last year than in the 20 years before. But the popularity of the "clones of old clones" hasn't waned one bit. If anything, it's clear that there are even fewer really popular genres now than ever before. While we should absolutely push the experimental edge, what we shouldn't forget is that people are comfortable with the familiar. The bigger the change you want to make, the more grounded in the familiar it needs to be in order for people to build a mental bridge from where they are to where you want to take them. That's as true for our game designs as it is for our game tools.

"One of the biggest mistakes I've made in product design is asking engineers, 'Can it be done?' Unless you're asking a first-class programmer, the question is useless. More specifically, responses fall into one of three categories: (Lousy programmer) 'Sure, that's no problem.' (Mediocre

programmer) 'Nope. Can't be done.' (First-class programmer) 'I could do it like this and it'll take two weeks. Or I could make a slight modification like this and it'll take five hours.'"
Tzvi Freeman, "Creating a Great Design Document," August 1997

Of course it can be done. That's a silly question. I don't know many programmers that don't believe they couldn't find a solution to a problem given enough time. The "first-class programmer" is not the one that can give a producer an off-the-cuff time estimate, though. A first-class programmer can only do that if they've already done it before. And most of the time, we're not solving exactly the same problem as we've solved before—there are always new constraints that make the problem very different—otherwise, we'd all still be using SNES game engines.

A first-class programmer will be able to tell you what they don't understand about the problem, what the risks are as they do understand them, and what they'll need to do in order to understand the problem enough to get a test model running. Or at the very least, figure out some way to scale the problem to fit in the time available.

"One of the most important things that I've learned in the past few years is that the number of hours you work does not necessarily correlate to your productivity. Productivity is based on the work that you accomplish, not the hours that you work."
Brian Hook, "So Long and Thanks for the Rail Gun," February 1998

Counting hours is as bad as counting lines of code. It's a nonsensical metric. But how do you measure real accomplishment, insight, and quality? How do you measure the things you didn't have to do because someone did it right in the first place? How do you measure the time that was saved because someone didn't make a brain-dead decision because they were too rushed to think it through? We haven't made a whole lot of provable progress on those metrics yet. But let's keep trying!

"Remember, however, that the quicker the motion, the greater the quality loss, because the area of newly exposed polygons (which are now carrying erroneous textures) will be larger. Teleportation, therefore, is problematic because all textures will be unexpected."
Jonathan Blow, "Implementing a Texture Caching System," April 1998

The techniques of texture caching and preloading haven't changed much, really. And the more general lesson is equally still applicable: You cannot properly design game systems without knowing the constraints of your (most relevant) data. A seemingly simple change—for example, enabling your character to teleport anywhere or move 10x as fast—can dramatically change the problem and therefore dramatically change the solution required.

There aren't many generic, one-size-fits-all solutions in the space of making games, and the all-too-common attempts to create them are a fool's errand. "Future-proof" is the perpetual motion machine of video game programming.

"We've got to bring entertainment to the places where most people socialize. I travel a lot. I see games in sports bars, but not in hotel lobbies. I see fast food locations without games. I see coffee bars with no games, and many chain restaurants with no games or amusements. In most of these locations, a pinball machine would look strange, but a countertop web terminal with easy touch-screen menus and gameplay would not."
Nolan Bushnell, "A Product Whose Time Has Come...Again," July 1997

I love that we have gone from a time not so long ago where there were so many places where games were not present, to games being present in virtually everyone's pocket. There is no place without games because people bring their games with them wherever they go. The final frontier for games is with our poorest kin, those who cannot afford the smartphones and tablets that many of us take for granted now. Looking forward to the next 20 years, I hope that everyone, everywhere will have games to play and to learn and to make, wherever they are.

"It's getting pretty bad for some guys out there. First, women wanted the remote. Then there was Title IX and that equal pay thing. But now it's getting serious, because we want your mouse, and we want it now... A man practically has to elbow his wife and daughters aside to get some quality time with his flight simulator these days."
Nancie S. Martin, "Take the Y Out of Computer Games," September 1997

Women had a tough time of it then. Women still have a tough time of it today. Not just women players, but women game developers. The difference between then and now, I think, is that some of us men are much more aware of it and would really like to do something to improve that situation. The industry as a whole can vary from "slightly biased" to "outright misogynistic," but it's never totally fair toward women.

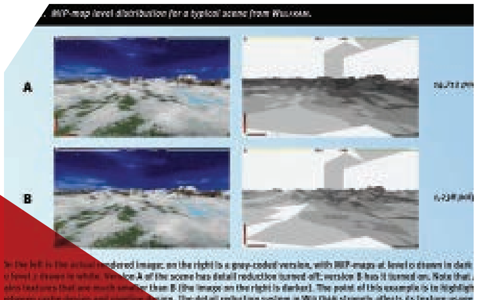
I'd like to see an industry that's fair, open, and inviting toward all people. Not for the sake of diversity, or for the different perspectives or ideas that could influence our games for the better, but because it's right. I'd like to see an industry where great women programmers aren't rare and a world where I get a chance to work with them.

"While you can pick up a book such as Bjarne Stroustrup's Design and Evolution of C++ to help you understand why C++ is such a screwed-up language, it still doesn't address the issue that C++ is a screwed-up language. It's constantly evolving, getting bigger and uglier, and pretty soon it's going to implode under its own weight."
Brian Hook, "How I Spent My Summer Vacation, or What I Learned While Working on Quake II," January 1998

Oh, boy. If anything, it's gotten so much worse. ip

***

Mike Acton is the engine director at Insomniac Games, Game Developer magazine advisor, and overlord of #AltDevBlogADay.


MIP-map level distribution for a typical scene from Wulcan.

On the left is the actual rendered image; on the right is a gray-coded version, with MIP-maps at level 0 drawn in dark ... level 3 drawn in white. Version A of the scene has detail reduction turned off; version B has it turned on. Note that ... also textures that are much smaller than B (the image on the right is darker). The point of this example is to highlight ...

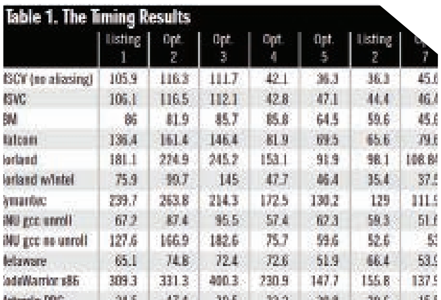
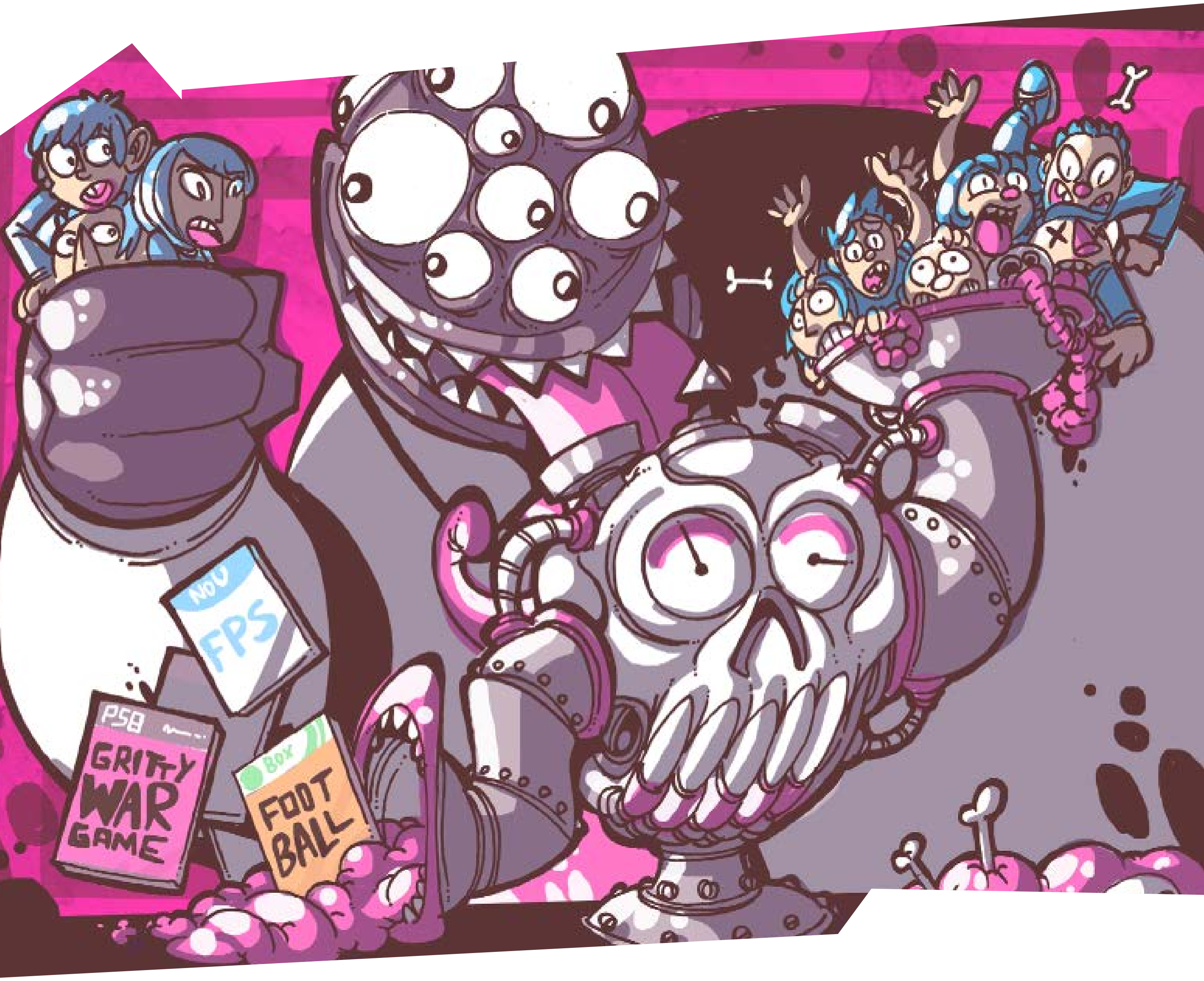


Table 1. The Timing Results

	Listing 1	Opt. 2	Opt. 3	Opt. 4	Opt. 5	Listing 2	Opt. 7
ISCV (no aliasing)	105.9	116.3	111.7	42.1	36.3	36.3	45.6
ISVC	106.1	116.5	112.1	42.8	47.1	44.4	46.4
BM	86	81.9	85.7	85.8	64.5	59.6	45.6
Watcom	136.4	161.4	146.4	81.9	69.5	65.6	79.6
Borland	181.1	274.9	245.2	153.1	91.9	98.1	108.84
Borland w/Intel	75.9	99.7	145	47.7	46.4	35.4	37.5
Symantec	239.7	263.8	214.3	172.5	130.2	129	111.5
GNU gcc unroll	67.2	87.4	95.5	57.4	62.3	59.3	51.6
GNU gcc no unroll	127.6	166.9	182.6	75.7	59.6	52.6	53
Metaware	65.1	74.8	72.4	72.6	51.9	66.4	53.5
CodeWarrior x86	309.3	331.3	400.3	230.9	147.7	155.8	137.5
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

THE LAST PIXEL

CLOSING THE BOOK ON PIXEL PUSHER



I've spent a lot of the last decade at Pixel Pusher repeating the mantra that game art is a business where nothing ever stays the same, and you need to constantly redefine yourself to stay relevant. This final printed issue of *Game Developer* is, sadly, a perfect example of the principle. Honing your craft as a purveyor of news and information printed on former trees is not, alas, the path to a prosperous future. Of course, the skills that go into writing a good article aren't dependent on the print medium (any more than core art skills are tied to a particular bit of software), so hopefully the astonishing blizzard of erudition, wit, and artistic insight that is Pixel Pusher will find a new home on the web. However, it's fitting that we spend our last few droplets of ink trying to make some sense out of the amazing and maddening things that have happened to our profession in the last decade.

When I started writing Pixel Pusher back in 2003, the column was primarily technical in nature. Most of the articles were how-tos and surveys of different techniques. This made sense at the time: Much of the toolset we take for granted today was just emerging. Today's staples, like subdivision modeling, blended animations, and programmable shaders, were all newsworthy back then. In those ancient days (MySpace debuted just a couple of months before I took over the column) the 'net did not offer a lot of specialty information for computer artists. However it soon became clear that the Internet was going to provide a better medium for doing tutorials and walkthroughs. Hyperlinks, multimedia, and downloadable support files are just hands down a better way to teach advanced

THE LAST PIXEL

graphics techniques than 2,000 words of print and a couple of images.

As that became clearer, the column evolved into a running meditation on what it means to be a game artist. We've looked at the question philosophically, artistically, economically, and technically—but after more than a decade of poking and prodding, I'm happy to admit that I don't have a pat answer. The games business—particularly our end of it—is completely and unabashedly mental. We try to hide it with solemn discussions about art and art history, with grand pronouncements about the future of media, or with high-tech wizardry. At the end of the day, though, we're people who make talking mushrooms, freaky deep-sea-diving cyborgs, and laser-armed zeppelins. Our profession is gleefully absurd even while it gropes for meaning and a chance to leave an imprint on the history of art. We combine the best and worst traits of adolescents: unfettered imagination and lack of common sense, incredible energy and ridiculous inefficiency, soaring passion and grumpy resignation. It's a puzzling life—but one we're very lucky to be living. Don't forget that part.

THE CHANGING OF THE ART

The craziness is a constant, but a lot of other things really have changed. We're a much more self-confident and ambitious bunch than we used to be. In 2004 I snarked about games that were naively hung up on the gimmickry of photorealism:

> "Somehow one doubts that kids are sitting around the cybercafes of Seoul saying things like 'Halo's subversive use of lens-flare radically deconstructs the notional game space.' So why does our industry devote such phenomenal energy to recreating the artifacts of other media?" (Pixel Pusher, May 2004)

Nowadays, of course, we've witnessed a great flowering of post-photorealist art. It would have been nearly impossible to get a publisher to listen to a pitch for games like JOURNEY or THE UNFINISHED SWAN a decade ago. Even triple-A titles like TEAM FORTRESS 2 and BORDERLANDS would have seemed too daring for the mainstream back then. But today we've overthrown the tyranny of photorealism so thoroughly that it's hard to remember the way the industry used to lap up grainy phototextures and choppy mo-cap in the name of "realism." At the other end of the spectrum, genres that used to be locked into big-eyed pastel trance colors have found different ways to tell stories too: Games like LITTLEBIGPLANET and LIMBO have shown how platforming games can build on the legacy of SUPER MARIO BROS. without being hobbled by it. In 2003, it would have been hard to imagine that all the fancy hardware and software of what we so quaintly called "next-gen" would be used to create VIVA PIÑATA. Even if you hate every one of these games—though if you do, you're crazy—it's a great time to be doing what we do: There are a lot more ways to sling pixels than ever before. It's okay not to give a damn about space marines or chain-mail bikinis or exquisitely accurate PanzerKampWagens. The tent is a lot bigger than it used to be.

While toting up the positives, it's also a good time if you've got the old-time artistic urge to Make a Statement(™). The "can games be art?" debate is wrapped up more conclusively than MASS EFFECT 3. There's endless, acrimonious debate about the nuances—the mixed critical reception of BIOSHOCK INFINITE is a case in point—but the mere existence of that debate proves the point: It's worth arguing about, therefore, it's important. Even the sentinels of high culture (most notably, the Smithsonian) have conceded the point that what we do occasionally rises to a level beyond peddling amusements. We knew that all along, of course:

> "As you read these words a kid somewhere is daydreaming about growing up to become just like a character you created; a group of friends is reminiscing about the great time they had visiting an environment you built; somebody is training their body to move in real life with the grace of an animation you created. People give a damn about what we do—sometimes for deep philosophical reasons, sometimes for complex personal reasons, and sometimes out of admiration for the dexterity and skill with which we do our jobs. That's what counts." (Pixel Pusher, March 2009)

But it's nice to have it down in writing.

PLAYING FOR REAL

Speaking of writing: There's a lot of it out there. Both reflecting and propelling the cultural acceptance of our art form, game criticism has escaped from the gravity well of commercial star-reviews and soared off into the ether in a way that would have been hard to predict a decade ago. That kid in Seoul is, in fact, probably talking (more likely tweeting) about deconstructing notional game spaces at this very moment. Bloggers, game journalists, and academics have arrived on our shores like conquistadors and are staking out claims left and right. You can buy(!) a 50,000(!)-word critical essay about SPEC OPS: THE LINE (*Killing is Harmless* by Brendan Keogh, which examines the entire game level by level with the intensity of a PhD thesis). We've finally achieved the most coveted distinction of artists everywhere: the chance to pick up a piece of commentary on our work and think, "What the hell is this joker talking about?"

All kidding aside, the rise of a literate and argumentative critical community is a great thing for us. We used to have a hard time seeing our own work in perspective when the only people whose tastes counted were commercial game reviewers, publishers, and marketroids cruising the bass-thumping halls of E3.

Like any commercial art form, we're always in danger of getting into ruts; sequelitis, copycatting, and creaky old tropes lurk around every corner. As criticism flourishes, it will teach us to see our own work with a fresh set of eyes, and help us be better artists as a result. Of course, just like fine artists, film makers, and authors, we'll alternate between fearing, despising, and desperately hoping to please the critics. For right now, however, take a moment to thank all those bloggers and academics who are trying to tell us how to do our jobs. The cacophony of critical voices out there will occasionally elevate your blood pressure, but it also helps keep you from getting stale.

CHURN OUT, BURN OUT

Now, if you've been working in this industry for most of the last decade, you've probably got enough reasons to get your blood pressure checked without any help from bloggers. The games business has weathered some pretty remarkable challenges over the last decade. We've gone from days when 30 people were a big studio to days when 300 wasn't considered unusual—and now we seem headed back toward 30 as casual, mobile, and indie games proliferate while triple-A leviathans founder (as this was going to press, EA announced another big round of layoffs in the hundreds). Along the way we've had to deal with flat salaries, outsourcing, and automation. Not to mention hardy perennials like crunch time and quality-of-life problems. Somebody out there really needs to take a look at the tuning on this thing—it sometimes seems like our game is stuck on Nightmare Mode.

There, alas, are long-standing reasons why so few people stick with this profession more than six or seven years: Work-life balance and lack of long-term career headroom were problems even at the height of the triple-A gold rush. Even on a 300-person team, there are only a couple of ladder rungs to climb. One of the most popular columns we ever ran in Pixel Pusher considered the fate of artists getting into their 30s and finding themselves adrift in midcareer dates from 2004:

> "...the games business has come to remind me of the glitzy shopping mall/utopia in Logan's Run. It's a fabulous playground for young people—though to be fair, the games biz is short on free love and polyester unitards—and we've all got blinking crystals in our palms, ticking away inexorably towards extinction. While we may not be facing the fiery Carousel

ILLUSTRATION: JUAN RAMIREZ

at 30, it seems like very few us stay in the business past 35." (Pixel Pusher, August 2004)

We revisited the same issue five years later, and thankfully the demographics had shifted a bit. By 2009, the average developer was between 31 and 35 (up from 25-30) and veterans were making noticeably more money than younger developers—which no doubt had something to do keeping artists in the industry past the five-year mark. It remains to be seen, of course, if that trend continues. Perhaps the vets will get a chance to relive their low-polygon glory days on mobile platforms: An iPad 2 has about the same horsepower as a PlayStation 2 did back in the day. On the other hand, smaller budgets and tighter margins might drag us back to the days of churning and burning through 20-somethings.

EVERYONE A GAME DEVELOPER

The graying of the industry creeps along slowly year by year, but other demographic changes are happening with amazing speed. The industry remains overwhelmingly white and male (and, dare we say, slightly pudgy?) but that's changing fast. Everybody is a gamer—hell, my parents play games nowadays—so it's not surprising that everybodies of all backgrounds want to make games too. However, this change isn't going to be seamless; the fervor of last year's #1ReasonWhy campaign shows that we've done a pretty poor job opening up our profession. Unfortunately, it also shows how easily efforts to change that can evoke defensiveness and derision. But let's face it: It's literally inevitable that the cozy monoculture most of us learned our trade in is going away. As the industry broadens its base, it's also going to be more loosely knit and fractious.

That's actually a vital thing for our medium. Differences of opinion and point of view keep us from getting boring and repeating ourselves—something we've been, ahem, occasionally accused of. However, it's also going to change the way we relate to our workplaces and our profession. We need to embrace the new reality with the same adaptability that we show when our professional skills get obsoleted every few years. We're good at constant relearning, and this will be an excellent opportunity.

GAME OVER

Which brings us back to where we began. Nothing lasts forever: not DPaint, not Character Studio, not the MMO boom, or even the EA football monopoly. And not, alas, the print version of *Game Developer*. On the way out the door, let me offer a profound thanks to all the folks who've read and responded to the column over the last decade: The game art community is an amazing place, and it's been a privilege. Thanks, too, to all the good folks at the magazine who made it all work. Nothing lasts forever—but in a world where XCOM can come roaring back to waste as much of my life in 2013 as it did in 1994, who knows what the future holds? f

***

Steve Theodore has been pushing pixels for more than a dozen years. His credits include MECH COMMANDER, HALF-LIFE, TEAM FORTRESS, COUNTER-STRIKE, AND HALO 3. He's been a modeler, animator, and technical artist, as well as a frequent speaker at industry conferences. He's currently the technical art director at Seattle's Undead Labs.

PLUS ÇA CHANGE...

OPPORTUNITIES FROM ENDINGS

KIM PALLISTER

For the past two years, Dave Edery and I have tag-teamed *Game Developer*'s business column. We've covered a wide variety of topics ranging from crowdsourcing, to free-to-play, to copyright issues. If there's one idea common to most of them, it's perhaps that of disruption and change in the industry. So while we are sad to see the end of *Game Developer,* it's only fair to note that there's a tinge of irony to the fact that it's being brought about, in part anyway, by the same types of disruptive forces.

Magazines are an information vehicle that makes less sense than it used to, because the way in which readers learn and share information has changed. Blogs, mailing lists, and Twitter subscriptions keep people up on the latest news. Community sites make it easy to find technical solutions *and* give quick access to the authors for feedback. Even the related wisdom-of-crowds and decentralization is challenging the top-down structure of how information is disseminated—from wikis to un-conferences. But while the form changes, the theme remains constant: We are learning from one another. The game industry has always amazed me at how open and sharing it is in most of its disciplines. Deep down, everyone really just wants to see people make great games.

SHARE YOUR IDEAS.

Whether it's in a pressed collection of tree pulp, a blog post, an impromptu talk at an un-conference, or some augmented-reality document posting on the Moscone walls at GDC 2018, the value of these contributions is a constant, and a key part of what keeps the medium moving, regardless of the vehicle. *Plus ça change*, and so on. In the meantime, I'll look forward to this last hard copy issue coming out, and store it next to my first issue from so many years ago as a chapter of our history.

DAVID EDERY

I remember when I first landed the job of worldwide portfolio manager for Xbox LIVE Arcade back in 2006. It felt much like winning the lottery—I had no game industry experience, no "portfolio management" experience of any kind, and no particularly strong professional references. And for the first year or so, despite some rather insane moments, it truly was a dream job. The next couple of years weren't so hot.

As it turns out, I have neither the skill nor the temperament to survive in a highly political environment. This is not a knock on Microsoft; the vast majority of large companies are highly political entities. Only someone naive or foolish seeks a highly visible business role in a huge corporation without being willing to "play the game." That was me.

So I allowed myself to be swept passively from new manager to new manager, in one reorganization after another. My title never changed but my responsibilities slowly atrophied. Despite this, I was comfortable; I enjoyed the perks of my role too much. So I allowed myself to rot in a job that was clearly no longer right for me (nor me for the job) until Microsoft did the merciful thing and laid me off along with thousands of other employees in 2009.

Anyone who has ever been laid off knows what this feels like. No matter how much you dislike your job, there's still a small part of you that can't help but feel ashamed when you get laid off. You don't look forward to your next conversation with any of your co-workers (or with your spouse, for that matter). Which makes it all the more notable is that being laid off by Microsoft was one of the best things that ever happened to me.

Being laid off forced me to pull my head out of the sand and ask myself what I really wanted to be doing with my life. There were once-in-a-lifetime opportunities exploding all around me: f2p games, mobile platforms, social networks—and I had been letting them pass me by. So I started doing some consulting to pay the bills and to possibly lay the groundwork for a bootstrapped studio. And not long thereafter, I co-founded Spry Fox with my friend Daniel Cook. Spry Fox wouldn't exist if Microsoft had not hired me in the first place—or if they had never laid me off.

The death of something we care about is painful, but there is no growth or evolution without death. So let's bow our heads, pay our respects, and then give three cheers for the death of bad old jobs, irrelevant platforms, stagnant companies, and, yes, even lovely paper-based news products. The best way to honor the dead is to remember the good things they represented, and to put all your energy into the better things that come next. b

***

Kim Pallister works at Intel doing game industry forecasting and requirements planning. When not prepping the world for super-cool hardware, he blogs at www.kimpallister.com. His views in this column are his and do not reflect those of his employer.

***

David Edery is the CEO of Spry Fox and has worked on games such as REALM OF THE MAD GOD, STEAMBIRDS, and TRIPLE TOWN. Prior to founding Spry Fox, David was the worldwide games portfolio manager for Xbox Live Arcade.

IN THE END, TELL THE TRUTH

GAME ENDINGS AS A MOMENT FOR HONEST DESIGN

Argument: As a game designer, you are more free when crafting your ending than you are for any other piece of your game.

First of all, having an ending at all is your choice. Don't want one? All good! Games are loops, and if you want to leave yours closed, you will be in good company. No one has ever "finished" poker, or football.

But for games that do have an ending, only a small portion of your players will ever see it. We are, as an industry and as a culture, still confused about this. We are dismayed at the low finish rates of our games, and a player who puts down the controller before reaching the end is left with a vague sense of having dissed the game team.

Yet, the ability for players to stop playing whenever they feel like it is inherent in the form! This is not a bad thing; this is a good thing. It is part of the game-design landscape. And if you learn to worry less about insisting that everyone who starts finishes, and put your attention on the advantages this fact of gaming gives you, you will not find a more personally liberating moment in game design than in designing your end.

The question is: How will you use that freedom?

LEARNING TO DABBLE

For several years back in the late 1990s, I lived with an eccentric friend named Dylan. Dylan was a carouser, a lover of swords and theatrics, a collector of experiences—and an avid video game starter.

Dylan played dozens, maybe hundreds of games per year, and this was before the Internet, so they mostly came from the store. But, for all his passion, I don't know that I ever saw him put more than an hour into a single one. He would buy them, try them, love them... and then set them aside forever. This was a man who stopped playing DIABLO after an hour or so (!). Even more weirdly, he was always perfectly content with his purchases, never showing a single hint of regret at not seeing the end.

He never did this with movies or books. Ever.

Watching Dylan's weird relationship with the games he played taught me that it is absolutely not required to finish a game to appreciate it.

NOT A BAD THING

Last year, you may remember that CNN published an article by Blake Snow that regaled the Internet with the news that only 10-20% of gamers actually finish the games they started (http://bit.ly/q1ezhV).

No argument. When we see game finish rates over 30-40%, we sing the praises of the team and pop the bubbly. Numbers like that imply that we managed to make some seriously compelling content, and smooth out all the bumps along the way. Precious few games reach that goal.

But, I have a beef with an unspoken assumption in this article, and in many articles like it. Here's how the article's author put it:

"Let [this] sink in for a minute: Of every 10 people who started playing the consensus 'Game of the Year,' [RED DEAD REVOLVER] only one of them finished it. How is that? Shouldn't such a high-rated game keep people engaged? Or have player attention spans reached a breaking point? ...Who's to blame: The developer or the player? Or maybe it's our culture?"

My beef is with the idea that failing to finish a game is a bad thing.

Putting down the controller somewhere before the final climactic scene in a video game is not a sin. It is an intrinsic part of our art form.

UNFINISHED

I never finished the first BIOSHOCK, yet it remains a game I thoroughly enjoyed. GRIM FANDANGO? Never finished it. But I sure as hell use it as an example in design discussions! I have never finished a single Z, but, man, they are fun (usually).

There are a ton of games that don't even have endings. Most arcade-style games and most MMOs don't have real endings. THE SIMS doesn't have an ending. Poker? Chess? Football?

In fact, a broad majority of the world's long-standing favorite games are specifically designed to never be finished. One game of Sudoku leads to another, which leads to another...

In game design terms, even putting an "ending" into your game is, clearly, optional. We know this. It's self-evident. So, then, why do we gnash our teeth and tear out our hair when only 20% of players reach the end of our (story) games?

NOT A MOVIE

I believe that the idea has its roots in our beliefs about other media. There is an implicit rejection that is present when someone walks out of a movie, turns off a show on TV, or sets down a book unfinished. For those mediums, the message of this action is clear: "I'm not enjoying this story enough to continue."

When someone stops playing a game, however, the possibilities are far, far more varied:

"I'd love to keep playing, but the time commitment is too high for me."

"I enjoyed the beginning, but now it's getting sort of grindy, and that's not for me."

"Love the game, but I'm weary of the player culture, so I'm going to hang out somewhere else."

"My friends stopped playing."

These are not necessarily sins of the designer. Gaming is as much a lifestyle as it is entertainment, and if a game doesn't fit into an individual's life, they are going to put it down. That's not a tragedy. That's a feature of our design landscape.

So, instead of looking guiltily at our completion rates and fantasizing about a world in which 99% of the players who start our (story) game reach the final scene, let's flip it around and see what we can do to take advantage of this fact, instead.

FLIP IT AROUND

More than half of your players are not going to finish. You know that going in, so think of it as a design constraint! What does that mean to you?

First: The deeper into your game your content is, the more likely it is that the players that are still with you have been having a good time. They're in. They've bought it. You have earned a certain amount of faith capital with them, and they probably want to see what else you've got up your sleeve.

Second: Because your producers and various high-mucky-mucks have seen the finishing stats for other games, they know that dev time spent in detailed iteration on your ending is effort going to a small subset of players. They will prioritize the team's time accordingly. They will thus be more likely, whether through disinterest or lack of time, to let your crazy idea for the end slip through the cracks.

Third: Players themselves already know that arriving at the end is a rare occasion—because they, personally, most likely don't do it very often. Every player has put down the controller on at least a few games. If they do decide to complete the whole thing, they will wear that fact as a badge of honor (we hope). So, they are psychologically primed to receive some kind of acknowledgment for their effort. Bright-eyed, with the end in sight, your players look to the designer expectantly, ready to interpret whatever you present as a kind of reward, while your producers turn a blind eye…

TELL THE TRUTH

I only have one piece of real advice for you about this moment: Tell the fucking truth.

Whatever it is that is in your heart, whatever it is that has drawn you into making this game in the first place, do that with your faith capital. Spend it telling them that, somehow.

The first MODERN WARFARE had a great example of this: The final mission was the most over-the-top crazy, punishing, nearly-impossible-to-complete madness-fest in their game. It had almost no explanation, required none ("PLANE! TERRORISTS!"), and it was simply brilliant. The level was a celebration of the game that you had just finished, a self-referential guns-blazing cherry on the cake that was completely unnecessary, but became legendary.

One of the most satisfying endings I have ever played was the ending of THE DARKNESS. It laid bare the truth of the fantasy they had created, and gave me full rights to punish an evil that I had come to loathe. The truth there was consistent with the story, but it was the play that they created that made that last scene true. I hated the villain of that game, and in the end the game did nothing to force my hand (beyond closing the door behind me). When I took my revenge, it was me that did it, and that act stayed with me.

But it is the ending of the first METROID, perhaps, that best demonstrates the strange liberty we have with this moment. It could have ended with Samus Aran raising a blaster into the air in victory. That would have been satisfying, and it was an amazing game all the way through. Hero pose! Instead, Samus stepped out of the battle suit, demonstrated her gender, and shattered the 8-bit preconceptions of players everywhere. It is still one of the most celebrated endings in gaming history.

AS AN EXAMPLE

Let's say we were to apply these principles to this article.

You've stuck with me this far, so I can perhaps assume that you're interested in what I've had to say so far. We're near the end, so you are maybe starting to think about what you'll read next, or putting down the magazine. Perhaps you are looking forward to the internal satisfactory tick-mark that comes from reading the last line.

How might I use this receptive state of mind? What is my truth about endings, right now?

GD MAG

Speaking of endings, did you know that this is the final issue of this here magazine? Funny story: Through random luck, I've ended up with the honor of writing the final Design of the Times. That's this article, right here.

You know, the first time I picked up an issue of GD Mag was back in 1996, in the offices of Hyperbole Studios. I was a late-20-something, blown away to be suddenly making games after long years of professional wandering.

It was the existence of this magazine that gave me my first glimpse into the murky, somewhat-secret society of game developers. The magazine's professional-looking cover and its interior pages full of post-mortems and dev tricks all were clearly aimed specifically at a readership made up of people who made video games. Flipping through the pages, I gradually discovered that I very much wanted to be part of that target market.

It's much later now. We have internets, game developers are meeting with vice presidents, and 99.9% of people under 25 have played video games. It's a world in transition, and I cannot wait to see what happens next. But I, for one, won't move forward into that future without first pausing and, maybe just for a moment, placing an affectionate hand on the magazine that was the warm face that greeted me as I entered this industry.

Thanks. Thanks for that, and for all the other stuff.

ENDINGS SET YOU FREE

That is my truth on endings: I mark them, I use them to reflect, and if I can get away with it, I give thanks to people who have had an impact on my life.

As a game designer, you are more free when crafting your ending than you are in any other piece of your game. So, in the end, tell the fucking truth. Tell as much of it as you can manage. Tell it as best you can. And see if you can give the world something to remember. d

***

Jason VandenBerghe is a creative director at Ubisoft, which he has to admit doesn't exactly suck. You can read his intermittent blog and various scribblings at www.darklorde.com. He can be reached by email at jason.vandenberghe@ubisoft.com.

OUR INTERACTIVE AUDIO FUTURE

MODELING THE FUTURE OF GAME AUDIO

The following is an excerpt from the *Oxford Handbook of Interactive Audio* to be released in 2014.

MILESTONE 1: POWER ON

I'm standing on the bank of a river. It's late, and the heat of the afternoon sun is fading. The sounds of cicadas are all around me. Their placid chirruping, accompanied by the nearby burbling of a stream, sets a tone of peaceful tranquility. The simulation here is really good.

As I kneel next to the shore, splashing water on my face, the sound erupts in a perfectly modeled and synchronized symphony of harmonic fluids. Each drop of water from my fingertips that ripples on the surface is reflected in automatic sound synthesis and tied so tightly to the visuals that I'm struck by the fact that it was just a short time ago that we were locked into a sample-based methodology. It didn't happen overnight, but as I am standing on the shore of this digital river, it seems clear to me that audio technology has finally made good on the promises of transparent interactivity.

It should be said that I'm inputting this log in real-time, while working inside the beta release of our current simulation based on a historical re-creation of Earth circa 2012. The simulation has been developed using an industry-standard authoring application by specialists spread out across the galaxy. I'm currently reviewing the work done by our technical sound synthesis specialists who have recently updated some old sound models. It seems like ages ago that we began working with our library of reference materials to synthesize different aspects of Earth's soundscape into the living, breathing, and sounding world that is represented today.

MILESTONE 2: SOFT FOCUS

I remain lost in thought as the sound of rushing water catches my memory. My mind is transported to a sunny day from my past. A reunion has brought family members together within a branch of an older beta version of a simulation located by a nearby stream in the countryside of Earth 2012. It was during this time in simulation technology that our industry was just beginning to iron out inconsistencies inherent within the burgeoning field of procedural audio, synthesis, and the advanced manipulation of dynamic sound: baby steps toward the expansive fully realized simulation I'm testing today.

As we laugh and carry on inside the memory of my mind's eye, the children play and chase butterflies along the edge of the rushing water. From the corner of my eye I can see my young daughter rushing in and out of cattails twice her height with her favorite doll. Her look of freedom and wild abandon while chased by cousins as the fronds whoosh back and forth brings a smile to my face. I move to speak and urge her to stay clear of the undulating, black, and treacherous stream edge, but they are gone before the words leave my lips.

I become lost in debate with various relations over the use of sound as a storytelling device. It seems we have finally found a way to effectively use all five senses to convey an emotional arc as part of an interactive experience. Of course, there continue to be new ways to channel and leverage our history through simulations, new ways of combining what has already happened with new technologies as a way of making sense of the future. Through the lens of creativity, we may finally understand who we are and where our civilization is headed.

Back at the reunion, the sun is beginning to set in the distance, and it's time for us to take leave of this place. I head toward the pack of children collapsed by the riverside in search of my daughter. My steps echo coldly in the grass. Her smiling face goes unseen, so I ask the gathered assortment of nieces and nephews where she might be. No one has seen her for some time, they say.

MILESTONE 3: DEBUGGING MEMORIES

In a moment, I feel my every fiber stiffen toward a heightened awareness of my surroundings. Nothing has changed, and yet in that instant it's as if all sound had been removed but for the deep churning of water. I look around frantically, my eyes darting between reeds as I began calling her name. People are gathered around me, I can see their mouths moving but can no longer hear their questions. My mind is sharply tuned to the sound of dramatically frothing whitecaps as I desperately attempt to keep my thoughts above water.

I wade into the sea of cattails, moving further and further upstream. Their fronds brush past me in a noiseless hush of extreme focus. I can hear others who have crossed a nearby bridge and begun combing the riverbank on the other side. My voice, becoming louder and more urgent, punctuating the ebb and flow of water to my left, sounds far away. A curious rocky outcropping comes into view, as the taste of saltwater reaches my lips. I see my daughter playing noiselessly with her doll in the shade of a dark black stone column. I call to her, but she doesn't look up. I quickly close the gap between us and suddenly recognize that the sound of the stream has disappeared.

My daughter becomes startled by the sound of my voice calling her name. She looks up as I rush to her and hold her in a deep embrace, my footsteps resonating with an alien quality. The sound, clothing brushing against her white dress, explodes within the vacuum of soundlessness I find myself enveloped in. I look into my daughter's eyes as she holds out a ring of flower stems that she's woven together. She looks at me and smiles. When asked by friends later she'll say, "I wasn't lost, I just found a quiet place to play." We emerge from the noiseless interstitial space, call off the search party, returning to the sound of the world around us.

These early simulations weren't perfect, but the feelings generated by them resonate just the same. The difficulty of authoring propagation for every explicit sound once meant that the process was subject to gaps in the handwoven audio fabric. Every inch was "painted" by hand with sounds. It's easy to see, looking back on the technology, how these "dead spots" could exist—these places where no extraneous sounds would reach. Thankfully, people continued to work toward accurately representing sound propagation within the simulation . These advances were, over time, able to leapfrog the manual process that had been bogging down the process . I have to say that the radical sound mix change brought on by my alerted state operated perfectly, in retrospect. It's hard to believe the leaps and bounds that technology has achieved, even during my short life span—especially as I have found myself deeply embedded in driving the modern state of the art in our re-creation of Earth.

MILESTONE 4: AUDIO BETA

I return to testing inside the current-day simulation. I stand up from splashing water on my face and dry my hands on coarse cotton pants. This elicits a similar soft explosion of sound as on that day so long ago. I'm reminded of how long it took us to get the subtlety of movement and material interaction to accurately reflect the complex dance of physical sound .Tying into the cloth simulation was the easy part. Once we had all of that data on our hands, it was purely a process of sculpture (that is, mostly subtractive). Sound models became more complex still with the addition of liquids and temperature… but at the end of the day, our flexible synthesis models stood up to the barrage of parameterization from each of the systems.

I step back onto the path and continue on to a nearby watermill. I'm reminded by each footfall how long it took to get these footsteps right. Everything is now handled inherently by the procedural models in place, which take all sorts of factors into account: mass, velocity, materials, viscosity, and a variety of other things. What began as a purely academic pursuit swiftly blossomed into a creative endeavor thanks to some early front-runners in the field. Once the underlying mathematics were in place, the tools quickly followed. This development allowed early designers of sound to move beyond hard science running behind the scenes. Enabled by smart and creative authoring applications, the sound of footsteps and movement was able to transcend the synthetic quality of early experiments and emerge as a realistic representation.

As I approach, I note the churning sound of water on the undershot-wheel paddles grows louder with each step. I'm struck by the quality of dynamic frequency filtering over distance. The simple 3D point sourced, volume-only attenuation curves of yesteryear have long ago been replaced by a complex matrix of real-time distance modeling, taking into account everything from air density to temperature and humidity. The spinning wheel splashes on effortlessly while some unseen turbine labors to convert the power of nature into energy. Of course, this one is just for show. However, it wouldn't surprise me to find that it powers all of the lights inside. In such a complex woven fabric of interaction, you can usually count on everything being designed with a purpose.

I walk to the door of the water mill and reach for the handle with trepidation; the last time I tested it, it didn't accurately convey the weight and material type, so I sent the sound designer back to the drawing board. Turning the knob I hear the rusty mechanism come to life with a sharp snap of metal, hollow-wooden resonance, and deep unsettling creak. Each individual aspect of the sound takes into account the compositional breakdown occurring: damp location (as evidenced by the deep rust flaking off each hinge), wood density and thickness, and the age and tenacity of the steel. The door closes with a deep thud that reverberates as I let it close behind me . That was much better, I think, as I send off a friendly note to the designer who is a million miles away. Within moments I've received a response which reads like a sigh of relief.

With so much to do, it's hard to get caught up on the little things. Thankfully, the work we've been doing on the underlying physical modeling propagates throughout the entire world. Imagine if we had to place sounds by hand for every river or door; the worlds we're creating are just too large for such a labor-intensive process. While we do try to take a handcrafted approach to sounds that require something unique, the expectation that every nook and cranny sound convincingly Earth-like in this simulation involves more than just a few sound files spread across the face of the planet.

MILESTONE 5: STRESS TEST

Back inside, there is the pervasive, low-end rumbling sound of rushing water coupled with an oppressive clattering. I'm standing in front of a succession of wooden gears locked together in a spiraling groove of perpetual motion. The hollow "thonking" sound they make as each tooth locks into place is a direct result of the latest wood models developed by the foremost modal synthesis historians . Piggybacking on their research into old technologies and material types has given us a satisfying level of detail to the rich characteristic of wood. With their research in place, we were able to further embellish the machinery using several creative tools running within the simulation.

The whole contraption is shuddering with an unbelievable rumble that I can feel in my gut. With a gesture, I engage the authoring tools, and an interface for interacting with the world appears in front of me. As it springs to life, ready for action, I quickly navigate the controls and begin slowing down the water flow outside in order to hear the change in sound.

As the internal mechanism begins to wind down, there are no artifacts in either the pitch or timbre as the gears begin slowing to a halt—this isn't a simple sound file representation. You see, the only remaining samples of an actual watermill were recorded at the low sample rate of 192 kHz/24 bit, but we were able to use feature extraction across a diverse sample set and mine relevant data from these recordings and use it to inform the creative application of various processes and models. These samples were critical, since we had never seen a working watermill in person, and ended up affecting the overall sound presentation.

As things grind to a halt, I notice a gentle whistling sound finding its way through the cracks in the thatched roof overhead. Wind was the first and easiest dynamic synthesis we could apply to these simulations—we could apply pitched and filtered noise of different colors in combination with reflectors and deflectors, both abstracted within authoring toolsets and programmatically based on geometric

AURAL FIXATION_Damian Kastbauer

representations within the different environments. This technology was very futuristic at the time, and what it lacked in "natural" sound factor, it made up for in its ability to be modified in real time, using parameters from the simulations. As the technology progressed, the randomness of nature swiftly took the place of consistently sloping pitch curves and unreal-sounding representations.

My footsteps echo convincingly on the hollow wooden floor as I resume my test and calibration procedures. I find the watermill synthesis and modeling holding up well under these extreme circumstances. I stop my testing just short of flooding the entire mill in order to listen to the resulting forces of nature on the tiny primitive structure. Safe within the confines of the simulation, the properties of the debug shard that instantiated when I enabled the authoring tools gave me a unique perspective on the resulting mayhem. Distanced from the confines of the simulated physical world within this unique instance, I'm free to run amok. I'll save the creative joy of destruction for another day. I start the river gently flowing again and exit the tiny building and continue my testing.

MILESTONE 5: CONTENT COMPLETE

Outside, the sky has blossomed into a majestic eruption of purple and pink at dusk. The late afternoon cicadas have all been replaced with a chorus of crickets and occasional bird chatter. It's been so long since we were restricted to a single looping audio file that I don't even notice how diverse the soundscape is. While there continues to be a place for field recording as the basis for building a library of granular models and anomalies, it's no longer possible to capture these sounds in their natural habitat. These sounds may have existed in nature a long time ago, but that time has since past. Luckily we have access to a wide variety of artifacts from the 21st century, including field notes, recordings, and skeletons.

Some of the environments initially proved difficult to re-create. Isolating elements amidst the noise of then-modern culture proved easy enough at first. When it became more difficult was when we began to note peculiar behavioral changes between recordings taken from different decades in many of the vocalizing birds and animals. As the rise of industrialized society began to take hold, so too did the sound of a new era of machinery and technology. These sounds become an inexorable part of the complex auditory fabric of the earth and, over time, completely modified the speech and frequency range of vocalizations wherever nature overlapped with industrialized society. The difficult parts then became understanding the complex interaction that developed over time, and finding ways to realistically represent the ensuing soundscape. The result is a blending of both natural and manufactured sounds in a complex cacophony quite unlike anything heard since.

I watch the sun slowly slipping behind the rolling hills off in the distance and I'm struck by the true beauty that the 2012 Earth embodies in our simulation. Everything is represented with a rhythm that resonates throughout, from the tall grass gently swaying in the breeze, to the water wheel working in ceaseless syncopation. It seems that in this moment, there was a balance between the elemental forces at work and the swiftly encroaching hands of progress. It's impossible not to judge the years that have transpired in the interim as tragic when faced with such beauty. We all hope that this experience can serve as a future roadmap for how to proceed as a society, now that the damage has been done.

MILESTONE 6: BUG CRUNCHING

Back to work, I quickly navigate a user-interface terminal and instantiate a new area of the world to test in. A transition opens soundlessly in front of me and floods my senses with the sound of the city. I'm overwhelmed by the density of the experience: all oppression and intensity as drones weave their way in and out of a sympathetic embrace. This humming metropolis is a hive, a thrumming and humming frenetic audio vibration that typified 2012 Earth in all its glory. In a moment: speeding cars, trickling fountains, and skyscrapers resonating in an orchestration of the then-modern age.

Between the insistent air-conditioner rattle and intermittent elevated train clatter lies the golden age of information technology, an endless stream of activity amidst the rapid acceleration of interconnectivity. It's not long, as I cross the threshold of modernity, before I'm confronted with the endless saturation of sensual information. These simulated molecules caress my battered brain-stem in a dance of orchestrated input toward an inevitable overflow of stimulation. Whereas moments ago I was adrift in the tranquility of the rural countryside, I'm now firmly frenetic in the no-hold-back-all-attack modern age.

I engage a frequency analyzer that immediately projects a rainbow of visual analysis across every corner of the world. I can use this debug model to visualize the density of the frequency spectrum based on color as it emanates from every object, radiating a kind of ethereal cloud of color based on the sounds of the city. As cars speed by I see their unique voicing characteristics reflected in a wash of frequency analysis, the hue and saturation accentuating a buildup of tonality across the spectrum. I'm looking for an anomaly reported by early adopters of this simulation, some sort of unnatural resonance.

Loading a restore point, the world around me snaps into its prerecorded routine. It must have been a familiar scene: buildings reaching to the sky, transportation of all kinds treading familiar paths, a swarm of people vibrating with energy. Vehicles zipping around like oversized insects buzzing with momentum, intent on reaching their destination through a complex mechanical ballet of belching exhaust and the endless drone of so-and-so revolutions per minute. Each building resonating with a hum. From the depths of their mechanical rooms to the vibrations of people cascading throughout the interior, each building is like an oscillator in this polyphonic synthesizer of a city.

MILESTONE 7: ANOMALY

It's the same on the sidewalks that flank every thoroughfare. Everybody on the street moving, seemingly in sync, toward an unseen destination. Each person moves lockstep in time with the heartbeat of the city; like a million tracks of kick drum mixed below the threshold of hearing, but undeniably propelling each step. The individuality of their concrete footfalls is lost in a sea of clear density .

Aside from the occasional amplitude increase, so far, the playback seems normal. I scrub through the capture to the moment specified in the report and all at once the world is saturated in a haze of pink, representing a frequency pileup. I switch to manual control and reverse to the origin of the anomaly. I peruse the data streaming in one of the debug windows in an attempt to isolate the source and immediately see the problem. I navigate to the source; it seems at first glance to be a discrepancy in the modeling. As I trace along a sinuous pulsating line that connects the extreme data-point to a mass of overlapping frequencies, I find myself standing low to the ground somewhere in the middle of the sidewalk.

I switch off the frequency analyzer and find myself face to face with a little girl sobbing uncontrollably. This explains it—the quality of our voice modeling always seems to break down during moments of extreme emotional response. Our models take into account the simulated size, shape, and composition of the simulated vocal chords and we've captured and re-created the entire range of motion, but there are extremes in every case that still require fine-tuning.

ILLUSTRATION: BRENT PORTER

MILESTONE 8: SYNTHESIZED SOUL

Of course we began by nailing down celebrity vocalizations first: These were the most profitable. As soon as an actress or actor achieved a certain status we rushed in to take a vocal snapshot which could be perpetuated beyond the peak of their profession. Coupled with a full body scan, procedural animations, and phoneme-based lip-syncing, we've had digital doubles as part of our experiences for a long time. There still exist moments when even our best technology cannot yet achieve the artistry present within a live body, though. At the end of the day it is still the performance, whether acted or authored, that has the potential to connect with the audience. It remains a testament to the human body-instrument that there are still secrets held within.

The girl seems to have quieted down and her face begins to scan the crowd frantically. My heart goes out to her as I remember that day so long ago when, as a parent, I lost someone dear to me, if only for a moment. Of course this is still only a simulation; none of this is happening, but I'm still here having a real reaction that I can feel in my chest as my breathing accelerates. It's more a result of my experience than any belief in this as "real." Regardless, I kneel down on the sidewalk in front of her and ask if she is lost. She looks in my face not knowing exactly what the answer is, but eventually nodding in acknowledgment. I begin to ask who she was with when, midsentence, the message is broken by a mother quickly lofting the little girl into her arms. Amidst a volley of hugs and exasperated condolences she's quickly whisked away through the bustling parade of lunchtime foot-traffic. I can see the little girl's smile through the sea of people as I return to the task at hand.

Scrubbing back in time to the original anomaly, now identified as the little girl's cries, I utilize the controls attached to the data-point for the girl's vocalizations. Soloing the sound of her voice, I begin to apply various smoothing algorithms. Without losing any of the ferocity or emotion, I methodically constrain the parameters that have caused things to become unnatural. The process takes just seconds, but in a simulation of this scope it could take hours to address every similar case. Instead, I apply the adjustments to the vocalization models to be used in the event of other such anomalies. While the changes don't go as deep as the simulation itself, they can be applied in real-time if the same behavior is found. If it happens often enough, the solution will be used to inform the next update to the vocalization model.

This period of Earth has become known as a "great turning point" in the evolution of life on the planet. People began to notice the changing environment. Even amidst the abstracted nature of the city, people were taking note of the fact that everyone had a role to play in preserving the planet. By the time this recognition spread, the focus had already shifted to worlds beyond the confines of a single planet. From there, it was a combination of exodus and slow decay.

I exhale and return to the task at hand, flipping to the next restore point that needs investigating. I'm faced with a wall of water, an undulating sea, and wind set to tear the roofs off the nearby houses. The howl and moan as the waves increase their amplitude approaching shore is unhinged in a moment of sheer sonic terror. I bring up my display, and prepare to orchestrate the power of nature.

MILESTONE 9: SHIP IT

Through a combination of fictional story and factual reference the idea of this story is to inspire the work being done to help envision the way forward for interactive audio. While some of these workflows and methodologies live strictly in the realm of science fiction, there are aspects that can be found running in simulations at universities today. This radical change in approach—from the standard sample playback methodology into a composite toolbox which incorporates extensible procedural, synthesis, and physical modeling techniques—is rapidly evolving within today's game industry toward a future hybrid model of dynamic sound. Due to increasing size, diversity, and complexity inherent in most games, it seems inevitable that sound needs cannot be met with sample-based content alone. I continue to look forward to the creative ways that sound can reinforce the perceived reality of an interactive experience by leveraging the inherent dynamism of simulations.

"Life's like a movie, make your own ending, keep believing, keep pretending." —The Muppets af

***

Damian Kastbauer is lost at LostChocolateLab.com and on Twitter @lostlab.

[1] Noriko Kurachi - Now Hear This: www.cgw.com/Publications/CGW/2010/Volume-33-Issue-9-October-2010-/Viewpoint.aspx

[2] Sounding Liquids - Automatic Sound Synthesis from Fluid Simulation: http://gamma.cs.unc.edu/SoundingLiquids/

[3] Efficient Numerical Simulation of Sound Propagation: http://gamma.cs.unc.edu/propagation/

[4] Precomputed Wave Simulation for Real-Time Sound Propagation of Dynamic Sources in Complex Scenes: http://gamma.cs.unc.edu/PrecompWaveSim/

[5] Motion-driven Concatenative Synthesis of Cloth Sounds: www.cs.cornell.edu/projects/Sound/cloth/

[6] Precomputed Acoustic Transfer: Output-sensitive, accurate sound generation for geometrically complex vibration sources http://graphics.cs.cmu.edu/projects/pat/

[7] RESound: Interactive Sound Rendering for Dynamic Virtual Environments: http://gamma.cs.unc.edu/Sound/RESound/

[8] Fast Modal Sounds with Scalable Frequency-Domain Synthesis: www-sop.inria.fr/reves/Basilic/2008/BDTVJ08/

[9] Walter Murch, Dense Clarity – Clear Density: http://transom.org/?page_id=7006

Brandon Sheffield

THE CHOICE

KNOWING WHEN TO FOLD

ILLUSTRATION: JUAN RAMIREZ

INSERT CREDIT_Brandon Sheffield

My day job at present is starving indie developer at Necrosoft Games—but in order to pay the bills, I often do side projects. (This column is one of those.) So it's in that capacity that I was doing freelance narrative design for a company in Asia.

It was going quite well, for a time. I solved some inherent problem with the story's structure, and solved some incredibly difficult questions, tying the whole thing together quite nicely. It took a lot of pacing in my friend's empty pool, and a lot of talking to myself, but it got there.

I then began work on the actual writing, because as any good narrative designer knows, the writing can't really begin until the structure of the game design is in place. In addition to general dialogue and the like, I also made a few side stories, which could be unlocked given certain conditions. Most players would never see them, but they'd be a nice bonus for people who wanted to get more out of the game.

These stories were written in a fairy-tale style, and were meant to evoke a mood of magic and whimsy. And it's one of these whimsical stories that brought the whole project crashing down.

You see, in one of these fairy tales, I discussed a character who began life as a boy, but whose mother always wanted a girl. He never felt right as a boy, and so dressed as a young lady with the things his mother had bought. But the other children made fun of him. Then, one day, through some magical occurrence, he was able to make a wish. He wished to "be normal," so the children wouldn't make fun of him—and in that wish being granted, he was turned into a girl. Then she was happy, and the children accepted her, because she was finally who she was supposed to be.

Now, in my mind, this wasn't a radical statement, and I didn't really think twice about it. But one person on the team, the art director specifically, took issue with it. As a Christian, he said, he couldn't abide something so immoral as changing gender.

Well, that basically tore the whole thing apart.

THE STANDOFF

If they'd criticized my writing, or said it wasn't interesting—that I could have accepted. But immoral? That I could not abide. After all, changing to one's proper gender (again, not that I was trying to make a point here!) is not a moral choice, it's one of necessity and is more about feeling comfortable with oneself than it is about any doctrine.

The art director, though, said he couldn't work on this game if this immoral thing were allowed. So the game's director, a friend of mine, appealed to me to make a change, while the creative director of the company, also a friend, appealed to the art director.

The game director just wanted the game finished. He didn't care either way about this particular issue. The creative director was nearly as offended as I was by the art director's statement, bringing "morality" into a story issue.

So here I was, with a hard choice to make. Ultimately, I was hired to take control of this part of the game's production, and the art director had chosen to poke his nose into my part of the process, likely because his staff had to do the art for this little story. But he was the art director for the whole company, and as a contractor, I have no leg to stand on there.

If I were flush with cash, I would've immediately just said "nope!" and moved on to the next project. But as I mentioned in the first line of this article, there's that "starving" element of the whole indie game developer thing. I could definitely use the money. So I actually had to think about this.

I asked my friends what they would do. Some said I should quit the project. Some said they admired the fact that I was even thinking about this as a choice, but that that side of the argument, the side that doesn't accept people for who they are, is slowly losing the battle in the grand scheme of things. So, I should suck it up and change this small bit of my story, and choose a bigger battle to fight next time.

But ultimately, my pride wouldn't let me back down. If I changed my story, then the art director would "win." He would get to feel justified in his moral stance, because I had backed down. I thought to myself: I can always get more money, but once I let my moral ground slide, that slide is forever. I'm now a person who has made that concession. If I've made that one, maybe I would make another.

Perhaps the art director felt the same way, and I almost respected him for that, as much as I disagreed with his stance. And so it came to be that I had to withdraw the entirety of my work from the game. It was all or nothing. I couldn't work on a game where skewed religious doctrines would dictate the game's direction.

The company viewed it as his fault for delaying production (by more than a month), but I certainly felt guilty myself. Should I have just sucked it up and let it go for the sake of the team? What price is personal honor and respectability?

THE HIGH PRICE OF DIGNITY

For me, ultimately, it wasn't a choice. I had put too much of myself into the game to let it be changed for a reason that seemed so silly to me, and wasn't based on performance or skill. On top of that, I had named the main character of the game after a close friend who passed away last year, and wrote the character as though it were her, meaning they'd have to change the name, as well.

A friend of mine said to me, "You've got to stop putting so much of yourself into your contract work!" And he's right. But when a project feels right, it seems like a good thing to do, to get personally invested, and make the game your own. It feels strange to step back, but perhaps it's for the best.

So I pose this question to you: Perhaps this battle wouldn't be one you'd fight. But what would you do when faced with a dilemma of this nature, using terms that prompt you to action? We've all heard stories about publishers wanting to change a character from black to white, or female to male, in order to "access the mainstream." How would you fight, in these situations? How have you fought?

I'd love to hear from you, though this is my final column for the magazine. Let's keep on trying to make our virtual worlds better places for everyone to play, and don't let the bastards get you down.

***

Brandon Sheffield is director of Oakland, California–based Necrosoft Games, and editor emeritus of Game Developer *magazine. He has worked on over a dozen titles, and is currently developing two small-team games for PlayStation Mobile. Follow him on Twitter via @necrosofty.*

FOCUS ON

Stardock Entertainment
15090 N. Beck Rd.
Plymouth, MI USA
Tel: 734-927-0677
Fax: 734-927-0678
jobs@stardock.com
Please use: Lead Game
Designer for your Subject
www.stardock.com

Stardock® ENTERTAINMENT

Lead Game Developer Opening

Stardock Entertainment is looking for a motivated and talented Lead Game Developer to join their team in Plymouth, Michigan. They offer an opportunity to lead the technical development of the company and to work with in a team-oriented, informal and flexible work environment. This position involves handling the technical aspects of games development and requires excellent multitasking, organizational, communication, and team collaboration skills. In this position you will lead, coach and grow a highly motivated and well balanced team. As the technical lead for the studio, you'll need to have expertise in:

- C++ coding and systems design
- Monitoring performance, memory usage, planning/executing performance and memory optimizations
- Monitoring stability, troubleshooting and resolving issues
- Designing, documenting and communicating system architecture
- Establishing and enforcing coding standards
- Maintaining a schedule and hitting deadlines

About Stardock

Stardock Entertainment has been developing and publishing games for over two decades. They are best known for developing the Galactic Civilizations series and publishing the Sins of a Solar Empire series. Stardock thrives on individual excellence, adaptability, and fun, while maintaining a structured environment.

requirements

- Experience leading a team of programmers
- At least 5 years of game industry experience (preferably with turn based strategy games)
- Expert C++ coding and systems design experience
- Excellent communication and mentoring skills
- Excellent organization and time management skills
- Excellent delegation skills

FOCUS ON

gamescom 2013: The entire gaming world in one place

The concept of the world's largest trade fair and event highlight for interactive games and entertainment is unique: it networks the entire value-added chain, from development and publishing to retail and the consumer. As the largest event of its kind in the world and the leading trade fair, it provides discussion platforms on all levels. It covers the entire spectrum of the international gaming scene:

- PC games
- Online games
- Video games
- Browser games
- Social games
- Mobile games
- Gaming hardware

The concept provides individual platforms for all target groups:

ENTERTAINMENT AREA: emotional gaming presentation for all – the world's largest playground for interactive entertainment
BUSINESS AREA: international meeting point for exhibitors, trade visitors and media representatives
GDC EUROPE: largest European developer conference

gamescom 2013 started with an exhibition space of 140.000 squaremeters. The trade fair and event highlight for interactive games and entertainment demonstrates with its leitmotif what the games world can expect in Cologne from 21st to 25th August 2013: The international games community – developers, providers, trade visitors, media representatives, retailers and thousands of gamers – meets at gamescom 2013 in order to experience together spectacular innovations and to celebrate the games and entertainment event of the year.

Koelnmesse and its partners, headed by the BIU (the German Trade Association of Interactive Entertainment Software), are already working flat out to further develop gamescom as Europe's central business and entertainment platform. gamescom awards are further developed due to the great popularity. The BIU also expects exciting novelties and innovations at gamescom when it comes to hardware and software innovations.

We look forward to welcoming you to gamescom 2013!

gamescom

Koelnmesse Inc.
8700 West Bryn Mawr Avenue
Suite 640 North, Chicago,
Illinois, 60631
Tel. +1 773 3269920
Fax +1 773 7140063
info@koelnmessenafta.com
www.gamescom-cologne.com

gamescom at a glance

- Business area, halls 4/5
- Entertainment area, halls 6-10
- Presentations of news and innovations of the entire industry
- gamescom award
- Games Developers Conference
- gamescom festival, City of Cologne

gamescom 2012 was a complete success

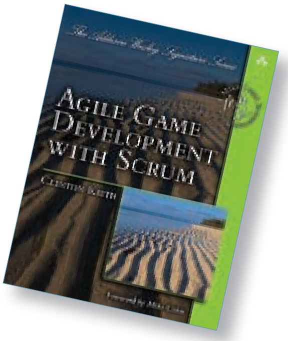
- 603 exhibitors from 40 countries
- 275.000 visitors in total
- 24,500 trade visitors (52% abroad)
- 5.300 journalists from 52 countries
- More than 120,000 additional visitors at City-Festival

CLINTON KEITH
CONSULTING

CERTIFIED Scrum Alliance
Scrum Trainer

Essential Scrum, Kanban and Agile for Creative Teams

Training by the author of "Agile Game Development with Scrum"


The Addison-Wesley Signature Series
Agile Game Development with Scrum
Clinton Keith
Foreword by Mike Cohn

Apple
EA
zynga
UBISOFT
id
LUCASARTS
Disney Online
ZeniMax ONLINE STUDIOS
NEXON
KIXEYE
HARMONIX
Autodesk
REMEDY
SONY COMPUTER ENTERTAINMENT
CCP
glu
BIOWARE A DIVISION OF EA
ACTIVISION BLIZZARD
WB GAMES
Tencent 腾讯

Learn how to apply agile practices such as Scrum and Kanban to creative products with training by Clinton Keith, the pioneer of implementing agile for video game development, the author of *"Agile Game Development with Scrum"* and a **Certified Scrum Trainer** and **Trained Kanban Coach**.

Get hands-on experience through team simulation exercises and instruction specifically tailored for creative product development professionals. Learn how agile applies to art and design as well as engineering. Discover how the collaborative team practices of Scrum and the visualization and flow management tools of Kanban can have a profound effect on your studio.

Agile/Scrum/Kanban are Proven to Help:

- Stop project crunch times
- Predict ship dates more effectively
- Control and reduce costs
- Energize teams
- Focus talent on creating value
- Eliminate barriers between art, design and engineering

Public and **customized in-studio** training courses teach iterative product development to creative teams and leaders at any level of experience. Gain practical knowledge through real-world examples and lessons from the development veteran and leader who introduced Scrum to the video game industry ten years ago. The courses apply simulation, discussion, improvisation and exercises to engage the creative developer and demonstrate, in action—not theory—how and why agile works.

Visit the website www.ClintonKeith.com, or email Clinton at clint@ClintonKeith.com to discover more about agile training and coaching options for creative teams.

"Clinton is a very switched on, genuine guy, and I strongly recommend his ScrumMaster Course to anyone who is looking to improve the productivity of their teams with Scrum." - *Kim Sellentin, Blizzard Entertainment*

"The training itself was first rate and received wide praise from our staff both for content and mode of delivery. Clinton's experience in game development and project management is a powerful supplement to his expertise in SCRUM methodologies. I strongly recommend both Clinton and his services." - *Michael Timothy Doyle, EA*

"He is a skilled instructor who brings passion and a tremendous depth of experience and know-how to his classes." - *Mike Cohn, Mountain Goat Software*

www.ClintonKeith.com **clint@ClintonKeith.com**

Who We Are

With more than 100 million registered players, InnoGames is one of the world´s leading developers and providers of online games. Our products (Tribal Wars, The West, Grepolis, Forge of Empires and others) are available in more than 30 languages. Our headquarter is located in Hamburg, Germany. We also have local offices in South Korea and Brazil.

The Business Principle

Our users have the option of playing InnoGames products completely free of charge and without restrictions for as long as they want. We also offer players the possibility of enjoying added benefits in the game by purchasing fee-based premium features. At the same time, InnoGames places a high priority on providing entertaining gameplay even without a premium pricetag.

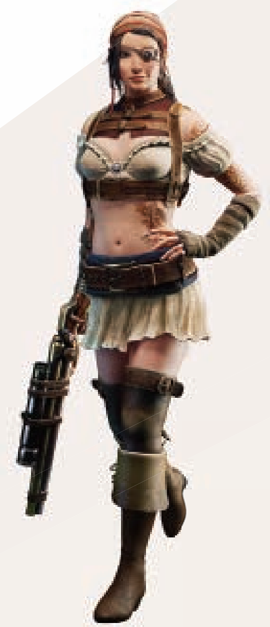
This principle has great advantages to the user. In contrast to traditional PC games, the product features here are already well-known before the player makes the decision to pay for added benefits - or not. The specific advantages offered by the premium features are also transparent. "What you see is what is what you get" is the underlying principle.

In our games, we focus on high long-term motivation. Our game Tribal Wars has been online for ten years now and continues to attract millions daily.

Partner with us

We have a good working network with trustworthy partners in all important game markets. As we are rapidly growing, we are looking for promising new partnerships. Are you a development studio experienced in mobile and browser games? Do you have high media competence and would like to cooperate with us? Or are you highly talented and want to boost our human resources? No matter your skill set - we are eagerly waiting for you. Just contact us

InnoGames GmbH
Harburger Schlossstr. 28
21079 Hamburg
Germany
+49 40 788 9335 0
info@innogames.de
www.innogames.com



We are looking for:

- Business Partners
  - High quality development studios
  - Effective media cooperations
  - Partners for our in-house affiliate program
- Talent to boost our human resources
  - Skilled mobile developers
  - Creative game designers
  - Artistic graphic designers
  - Experienced marketing, analytics and business development experts

Donya Labs AB
Gjuterigatan 5
582 73 Linköping, Sverige
www.simplygon.com

SIMPLYGON™

Automagic 3D Optimization

Simplygon is the leading tool-chain middleware for automatic optimization of 3D-game content and Level of Detail. Simplygon offers the benefits of LODs but without the drawbacks of tedious, time-consuming manual work; increased production time; and development cost. Moreover, using Simplygon means more consistent LOD quality; having LOD models available throughout/earlier in production; and increased visual quality by, for example, enabling more content to be shown on screen at the same time as well as minimizing LOD "pops".

Donya Labs, creators of Simplygon, is the leading developer and provider of automatic 3D-optimization solutions and is dedicated to helping professional game developers streamline their 3D-content pipeline. The revolutionary Simplygon middleware has been licensed by some of the world's most well renowned game publishers and developers.

Simplygon consists of 4 major Components

BoneLOD can be set to remove a specific number of bones for a LOD. Optionally, it can be run as a fully automated process, that removes bones, which does not influence the result much.

A unique and revolutionary technology, ProxyLOD generates a ultra-lightweight replacement mesh for one or more input meshes. This results in a drastic reduction in both poly-count and number of draw calls.

Further reduce draw calls using Simplygon's automated material retexturing and casting solution, to create texture atlases to replace multiple, complex materials.

Polygon reduction the Simplygon way. Our unique methods reduce triangles intelligently and the result is an automated process that guarantees a pixel-perfect LOD chain generation.

"We got a huge benefit from Simplygon on Tomb Raider. There was a critical point in the project where we had to make some sweeping concept changes to all of our enemy characters. ... We could have lost weeks of valuable production time late in the project, but with Simplygon we managed to condense the process into a few hours."
– Mike Abahazy, Lead Tech Artist, Crystal Dynamics

"Simplygon provides high-quality mesh reduction and time-saving benefits to both Epic and to the Unreal Engine community through our Integrated Partners Program, which brings the best features of leading middleware technologies to developers. Using Simplygon for our own games ensures that the tools are always meeting the Epic quality bar." *– Mark Rein, Vice President, Epic Games*

"The cost savings are enormous, should your game need LODs. A medium sized MMO can easily contain 7,000 models and a large MMO can contain up to 25,000 models if not more. If you estimate that it would take an artist four hours on average to generate several LODs per model, a small game will require 28,000 hours, or 14 man years. A large MMO would take 100,000 hours, or 50 man-years. In this sense, Simplygon can pay for itself over time."
– Bijan Forutanpour for GD magazine

THANK YOU *GAME DEVELOPER* MAGAZINE FOR ALL THESE AMAZING YEARS.

OBSIDIAN entertainment

Winner of multiple industry awards, including RPG of the Year for Star Wars: Knights of the Old Republic II, and Fallout: New Vegas, Obsidian is dedicated to the development and advancement of role-playing games for PC and console platforms. Obsidian was founded in 2003 by industry veterans Feargus Urquhart, Chris Avellone, Chris Parker, Darren Monahan, and Chris Jones who are best known for their work on critically acclaimed classics such as Fallout and Planescape: Torment.

Located in Irvine, California, Obsidian is minutes from beautiful beaches and just over an hour from the mountains and deserts. Irvine is home to excellent schools, wonderful housing communities, and countless bike and hiking trails. Boasting nearly year-round sunny weather, Orange County offers the tropical living you see in movies filmed in Hollywood, only Irvine sits on the safe side of the San Andreas fault - so when the big one hits, you and your family can explore the ruins of earthquake-decimated LA (less than an hour's drive away - not counting vault-dweller traffic).

Obsidian is looking for YOU!

Obsidian Entertainment is always looking for creative individuals interested in a challenging environment creating the best role-playing games in the world. If making the next generation of role-playing games in a friendly, enjoyable atmosphere is what you're looking for, we want to hear from you!

We're currently working on one of the most anticipated RPGs of 2013, South Park: The Stick of Truth, as well as the currently most funded video game on Kickstarter, Project Eternity. We continually iterate on our own suite of RPG dedicated tools and technology to evolve the RPG genre.

Obsidian Entertainment has the best of both worlds: cutting-edge projects of a large studio combined with the family feel of a smaller company. We emphasize employee development, while encouraging open communication both within and across projects.

Benefits at Obsidian

Being a part of one of the few remaining independent developers in the world creating role-playing games is only the start! Obsidian offers many benefits to its employees:

- Competitive salaries.
- Immediate 401k with matching and no vesting periods.
- Complete comprehensive employee health coverage, including vision and dental. Health coverage for family as well.
- Paid vacation, sick leave, holidays (including special company holidays) and an additional week off during the December holidays.

Apply

If you think you have the chops (that is, experience, qualifications and passionate desire) to create AAA role-playing games, send us a cover letter, resume, and samples of your work (portfolios and demo reels, sample code, personal websites or anything else that demonstrates your abilities) to:

jobs@obsidian.net

Obsidian Entertainment, Inc.
8105 Irvine Center Dr Ste 200
Irvine, CA 92168 USA

jobs@obsidian.com
www.obsidian.com

we are looking for

- Programmers
  - Mid-Sr Level Engine
  - Mid-Sr Level Gameplay
  - Mid-Sr Level Network
- Artists
  - Senior Environment Artist
  - Senior 3D Artist
  - Concept/UI Artist
  - Senior Technical Artist
- Designers
  - Level Designer
  - Systems Designer

CONTRACTOR CORNER

GDC VAULT HAS OVER 400 SESSIONS FROM GDC 2013 READY TO VIEW FOR VAULT MEMBERS ON ANY MOBILE DEVICE. DON'T MISS OUT, GET YOUR MEMBERSHIP TODAY!
GDC Vault
Featured Videos
Vault News

ADVERTISER INDEX

COMPANY NAME	PAGE #
ANIMATRIK	C2
ART BULLY PRODUCTIONS	62
CENTURION ART DEVELOPMENT	62
CLINTON KEITH CONSULTING	58
EA/GAMES LABEL	14
EA/POPCAP STUDIO	08
EPIC GAMES	C3
HAVOK	03
INNOGAMES GMBH	59
KOELNMESSE GMBH	57
OBSIDIAN ENTERTAINMENT	61
RAD GAME TOOLS	C4
SHARP SHADOW STUDIO	62
SIMPLYGON	60
STARDOCK ENTERTAINMENT	56
VANCOUVER FILM SCHOOL	23

IN ASSOCIATION WITH UBM TECH	PAGE #
APP DEVELOPERS CONFERENCE	38
GAMASUTRA JOBS	63
GDC EUROPE	26
GDC NEXT	33
GDC VAULT	62

For more information visit www.jointhegamenetwork.com

gd Game Developer (ISSN 1073-922X) is published monthly by UBM LLC, 303 Second Street, Suite 900 South, South Tower, San Francisco, CA 94107, (415) 947-6000. Please direct advertising and editorial inquiries to this address. Canadian Registered for GST as UBM LLC, GST No. R13288078, Customer No. 2116057, Agreement No. 40011901. SUBSCRIPTION RATES: Subscription rate for the U.S. is $49.95 for twelve issues. Countries outside the U.S. must be prepaid in U.S. funds drawn on a U.S. bank or via credit card. Canada/Mexico: $59.95; all other countries: $69.95 (issues shipped via air delivery). Periodical postage paid at San Francisco, CA and additional mailing offices. POSTMASTER: Send address changes to Game Developer, P.O. Box 1274, Skokie, IL 60076-8274. CUSTOMER SERVICE: For subscription orders and changes of address, call toll-free in the U.S. (800) 250-2429 or fax (847) 647-5972. All other countries call (1) (847) 647-5928 or fax (1) (847) 647-5972. Send payments to gd Game Developer, P.O. Box 1274, Skokie, IL 60076-8274. Call toll-free in the U.S./Canada (800) 444-4881 or fax (785) 838-7566. All other countries call (1) (785) 841-1631 or fax (1) (785) 841-2624. Vixen, who talks like that? (x6) Please remember to indicate gd Game Developer on any correspondence. All content, copyright gd Game Developer magazine/UBM LLC, unless otherwise indicated. Don't steal any of it. Or else.

M. WASTELAND & M. UNDERLAND

I'M LEAVING

A TAXONOMY OF FAREWELLS

UNRECOGNIZED LINCHPIN

Hi all, you probably didn't know this, but today is my last day. I didn't really get to meet many of you during the eight years I worked here. Possibly because I was the tester who sat in the server room most of the time.

Anyway, if you care, I'll be having a going-away thing at the bar down the street. You can come if you want. I was planning on having a beer and then heading home and watching *Babylon 5*.

P.S. Just a note. The build process might be a little slower after today. I used to fix 20 or so errors in the build manually every day at 5 a.m. so it could go out to the studio. Not sure if you guys knew that or not.

TRIUMPHANT EXECUTIVE

Loyal Employees,

When I joined this industry-leading company six weeks ago as your Senior Executive Vice President of Business Development, we were still in the nascent stages of solidifying our business model, brainstorming revenue streams, and seeking out disruptive influencers.

Now, operating profit in our planning spreadsheets has more than doubled, thanks to my innovative "double the numbers in the spreadsheets" initiative. And with our organization more ready than ever before to embark upon the long, hard road to profitability, I've decided to graciously step down in order to spend more time with my family and explore other opportunities. I am beyond proud to have paved the way for your future success!

Warmest Regards,

Brent Hornblower, Esq.
LinkedIn | Plaxo | Orkut

RAGE QUIT

Okay, this is the SIXTH TIME THIS MONTH that one of my burritos disappeared from the freezer, even though I put my name ON THE BOX. AND ON EACH BURRITO. I can't believe how many times I've told our producer about this. Heck, maybe he's the one eating them! You know, it's not like we're trying to get a stable build onto staging in time for the next update WHICH IS TOMORROW, and I need to EAT in order to be able to DO MY JOB. HELLO?

Actually, you know what? Screw this. I'M DONE. I'm out of here, like, for real. Let's see how well your burrito-stuffed faces can get that build up without me. Enjoy your FREE FOOD, provided by me to you for absolutely FREE, and get out of my life forever. PLEASE. GOOD RIDDANCE.

Someone forward this to HR.

P.S. NOBODY touch the mint-in-box action figures on my desk!!! I'll borrow my roommate's car and come pick them up on Monday.

GREENER PASTURES

Hey guys, I guess you heard by now that I'm leaving for that new place across town. Of course, I'm confident our games will be left in the best hands. That internal pitch for the new F2P title was particularly exciting—a summer camp theme! That sounds like it'll be super fun.

Anyway, I know things are tight after we didn't achieve the numbers we were supposed to with the last game, but it seems like the right lessons were learned. I'm kind of sad about leaving, especially now that the free gym memberships they've been talking about for a while might finally be coming through. My new studio apparently has its own gym on campus, but not to worry, I'll still come by for lunch once in a while for old times' sake.

Going to miss you guys, and really looking forward to seeing what this talented team will cook up next. =)

Bob

As some of you may know, Bob is no longer with the company. If Bob tries to email you, please do not read it and forward it to HR instead. If you see Bob in the parking lot, alert security immediately.

Nobody is allowed to talk about Bob.
That is all.

Thanks,
Management

AND NOW, A GENUINE GOODBYE

Dear Readers of *Game Developer* Magazine,

Thank you so much for reading our column over the last six-ish years. We had a great time writing it and the mail we got from you helped to sustain us through our own game development travails. Our favorite response was, "You must have worked at my studio!" And no, it was a different studio, but the patterns were the same.

To all line-level game developers out there, trying to do good work in a messed-up system: You are not alone.

Sincerely,

Matthew Wasteland, Magnus Underland

ad

***

Matthew Wasteland writes about games and game development on his blog, Magical Wasteland (www.magicalwasteland.com). Email him at mwasteland@gdmag.com. Magnus Underland writes about games and other topics at www.above49.ca. Email him at magnus.underland@gmail.com.

ILLUSTRATION: JUAN RAMIREZ

UNREAL ENGINE NEWS

BRINGING FEATURE-RICH GAMING EXPERIENCES TO THE WEB

How high-performance C++ games are running in HTML5 – no plug-ins, period!

The Web just became a much friendlier platform for beautiful, high-quality games thanks to the latest advances in Web technologies and Epic's ongoing collaboration with Mozilla.

Engineering teams at Mozilla and Epic Games recently ported Unreal Engine 3 to the Web in just four days using the powerful combination of asm.js, a highly optimized subset of JavaScript, and Emscripten, which enables developers to compile C++ code into JavaScript.

Epic then demonstrated *Unreal Tournament 3* in HTML5 at the Game Developers Conference and released *Epic Citadel* on the Web running in HTML5.

Developers can now achieve fast performance comparable to native. Bringing visually stunning, performance-intensive games to billions of people on the Web is now a reality.

Epic Citadel is built using standards-based technologies such as HTML5, WebGL and JavaScript, and should work in any standards-based browser implementing those features. The app's benchmarking mode has been a big hit with the community. Many users have posted screenshots with readouts of 150 fps and higher!

To try *Epic Citadel* for HTML5, download Firefox Nightly at nightly.mozilla.org and then visit unrealengine.com/html5 for a preview of the future of gaming on the Web. Instructions on how to unlock the frame rate are located within the FAQ.

EPIC GAMES

Come see Epic at upcoming industry events: **Electronic Entertainment Expo (E3)** (June 11-13, Los Angeles, CA), **Develop Conference** (July 9-11, Brighton, UK), **ChinaJoy** (July 25-27, Shanghai, China).
Email licensing@epicgames.com for appointments and sign up for our newsletter at unrealengine.com.